Project Cerberus
Firmware Update Specification

Author:
Christopher Weimer, Senior Firmware Engineer, Microsoft
Bryan Kelly, Principal Firmware Engineering Manager, Microsoft

Revision History

Date	Description
28/8/2017	Version 1.0
18/4/2018	Updates to the PFM section

Contents

Summary 7

1 Firmware Image Storage 7

2 Firmware Update 7

2.1 New Firmware Image 7

2.2 Update Active Image 7

2.3 Invalidating Signing Certificates 9

2.4 Firmware Update Process 9

3 Image Format and Verification 12

3.1 Boot-Time Verification 13

3.2 Image Update Verification 13

3.3 Application Launcher Format 14

3.4 Application Certificate Format 14

3.5 Main Application Format 14

3.6 Update Verification 15

4 Platform Firmware Manifest (PFM) Updates 19

4.1 PFM Management 19

4.1.1 PFM Activation 19

4.1.2 Validation Sequence 19

4.2 Send PFM Update 20

4.3 Update Active PFM 21

4.4 PFM Update Process 22

Table of Figures

Figure 1 Send New Firmware Image Command Flow 7
Figure 2 Image Update Command Flow 8
Figure 3 Firmware Update Process 10
Figure 4 Update with Certificate Revocation 11
Figure 5 Firmware Image Format 12
Figure 6 Boot-Time Verification Flow 13
Figure 7 Update Verification Flow 13
Figure 8 Firmware Image Verification 15
Figure 9 Application Launder Verification 16
Figure 10 Application Certificate Verification 17
Figure 11 Main Application Verification 18
Figure 12 Send New PFM Command Flow 20
Figure 13 Update PFM Command Flow 21
Figure 14 PFM Update Process 22

Table of Tables

Table 1 Application Certificate Format 14
Table 2 Main Application Format 14

Summary

Throughout this document, the term “Processor” refers to all Central Processing Unit (CPU), System On Chip (SOC), Micro Control Unit (MCU), and Microprocessor architectures. The document details the procedure for updating the Cerberus firmware image. It also covers the formatting and signing requirements for Cerberus firmware images.

1 Firmware Image Storage

The flash containing the Cerberus RoT firmware will be partitioned into three sections: Active, Recovery, and Staging. The active partition contains the current firmware image being used by the processor. The recovery partition maintains a known-good firmware image that can be used to restore the active image should something go wrong. The staging partition is the storage location for new firmware that has not been applied yet.

2 Firmware Update

2.1 New Firmware Image

Before any update process can be started, the new firmware image must be sent to the device. The image will be sent over the I2C bus after a trusted session has been established. The received image will be saved in the Staging area of the RoT flash memory, and any image information previously stored in this area of flash will be lost.

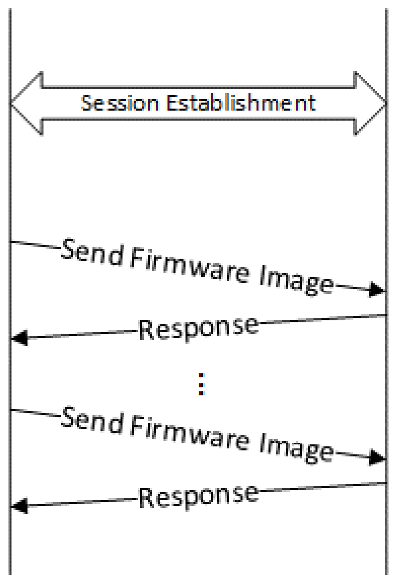


Figure 1 Send New Firmware Image Command Flow

2.2 Update Active Image

Once a new firmware image has been completely stored in the Staging area of flash, the device can be directed to use that image as the new running image. The update will only occur if the new image is verified to be a good. Verification of the new image is done using the same procedure as is used while booting the device. Once the image has been verified, it will be copied into the Active partition.

The response to the update command will be immediate and will only indicate if the request to start the update process has been processed correctly. To find out if the update is successful, the device must be polled for update status. The update status command will always report the state of the last update attempt until the device is reset or a new update request is received.

Once the update is complete, the RoT will start running the new image. Loading the new image will not be the same as a complete reset of the device. The current context of the system will be retained, so any active sessions will remain active. Also, boot-time initialization will not be re-run.

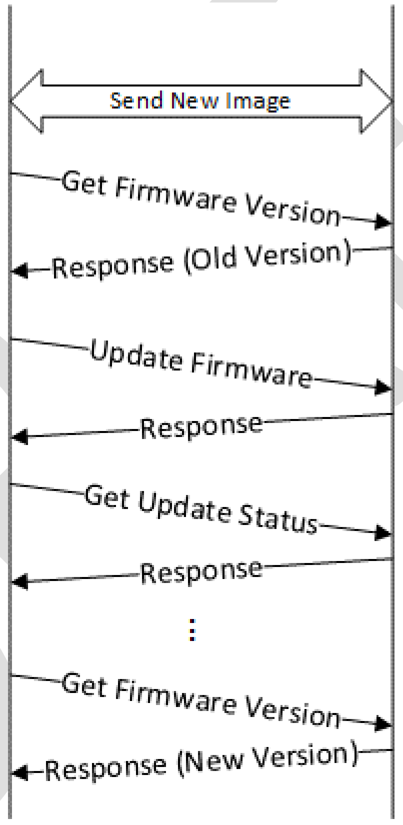


Figure 2 Image Update Command Flow

2.3 Invalidating Signing Certificates

When it is necessary to revoke a signing certificate, a firmware image must be created and signed with the new certificate. As part of the image metadata, it will indicate that the old certificate is no longer valid, and it will be revoked as part of the update process. A side-effect of revoking a certificate is that the recovery image will no longer be valid if it was signed with the revoked certificate. To ensure there is always an image that can be used for firmware recovery operations, the image in the Recovery partition will also be updated when the recovery image has been invalidated due to certificate revocation.

2.4 Firmware Update Process

The following diagrams describe the process executed by the RoT to update the firmware.

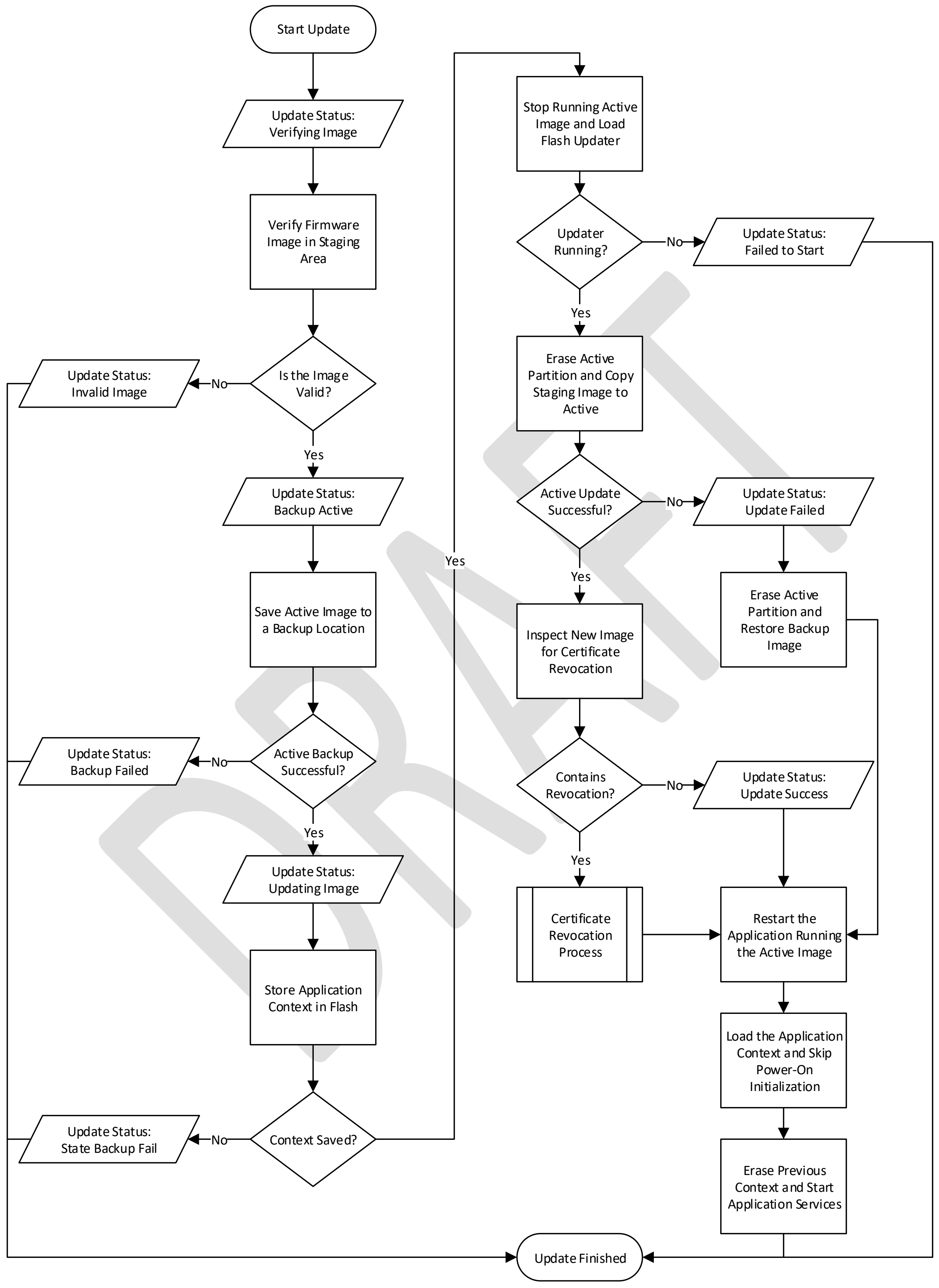

Start Update
Update Status: Verifying Image
Verify Firmware Image in Staging Area
Is the Image Valid?
No
Update Status: Invalid Image
Yes
Update Status: Backup Active
Save Active Image to a Backup Location
Active Backup Successful?
No
Update Status: Backup Failed
Yes
Update Status: Updating Image
Store Application Context in Flash
Context Saved?
No
Update Status: State Backup Fail
Yes
Stop Running Active Image and Load Flash Updater
Updater Running?
No
Update Status: Failed to Start
Yes
Erase Active Partition and Copy Staging Image to Active
Active Update Successful?
No
Update Status: Update Failed
Erase Active Partition and Restore Backup Image
Yes
Inspect New Image for Certificate Revocation
Contains Revocation?
No
Update Status: Update Success
Yes
Certificate Revocation Process
Restart the Application Running the Active Image
Load the Application Context and Skip Power-On Initialization
Erase Previous Context and Start Application Services
Update Finished

Figure 3 Firmware Update Process

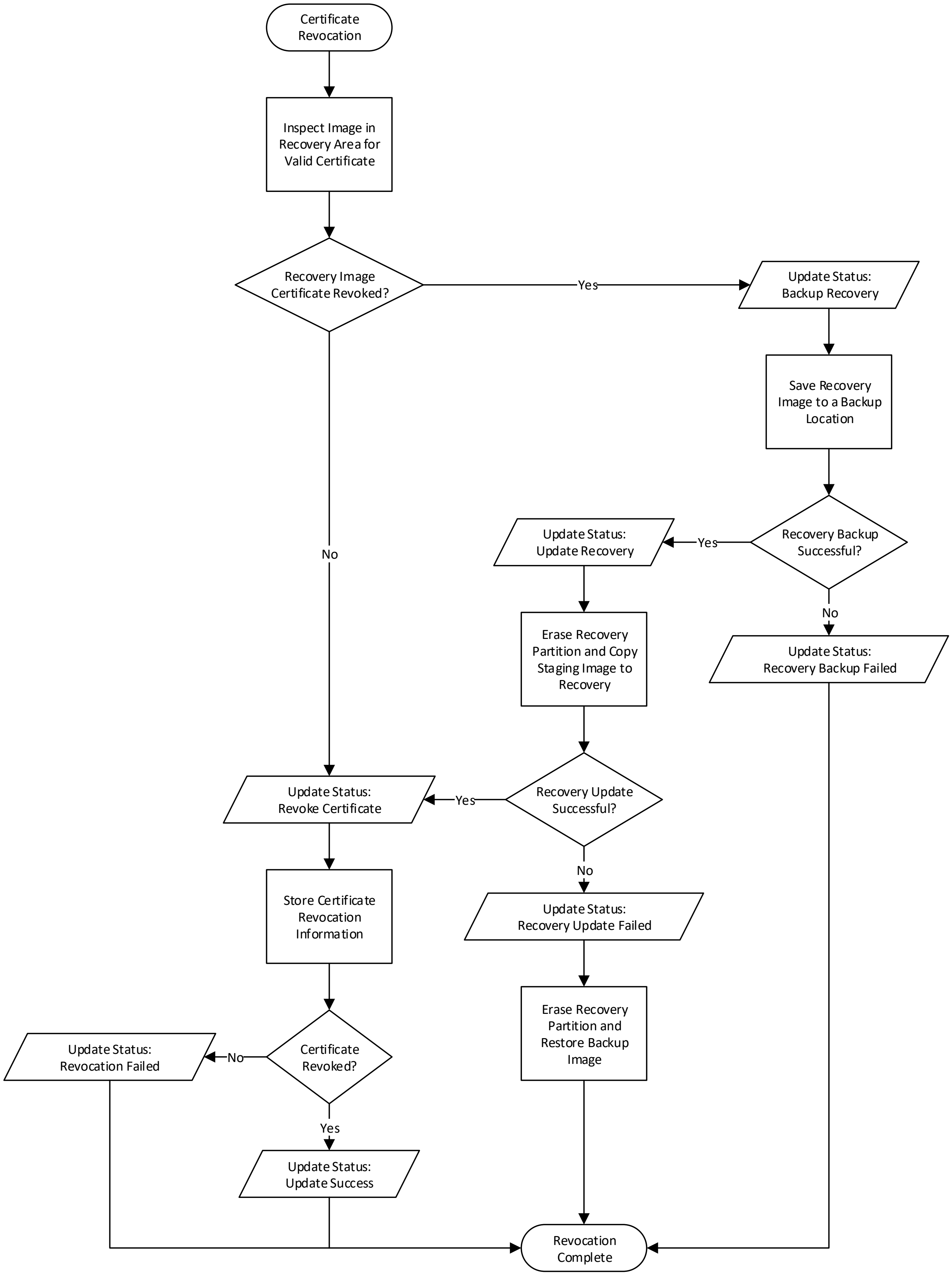

Certificate Revocation
Inspect Image in Recovery Area for Valid Certificate
Recovery Image Certificate Revoked?
Yes
Update Status: Backup Recovery
Save Recovery Image to a Backup Location
Recovery Backup Successful?
Yes
Update Status: Update Recovery
No
Update Status: Recovery Backup Failed
Erase Recovery Partition and Copy Staging Image to Recovery
No
Recovery Update Successful?
Yes
Update Status: Revoke Certificate
No
Update Status: Recovery Update Failed
Store Certificate Revocation Information
Erase Recovery Partition and Restore Backup Image
Certificate Revoked?
No
Update Status: Revocation Failed
Yes
Update Status: Update Success
Revocation Complete

Figure 4 Update with Certificate Revocation

3 Image Format and Verification

Both at boot and prior to running a firmware update, the validity of an image must be verified. In both cases, the same firmware components will get validated. The firmware image contains three different sections of data to facilitate this verification process. The first is the application launcher and firmware updater which will get validated using the certificate programmed into the device. The second is the certificate that is used to validate the main application image. This certificate is also validated using the root certificate in the device. The last part in the main application image, and is signed with the application certificate. This certificate will be checked to see if it has been revoked, and if not, will be used to validate the main application.

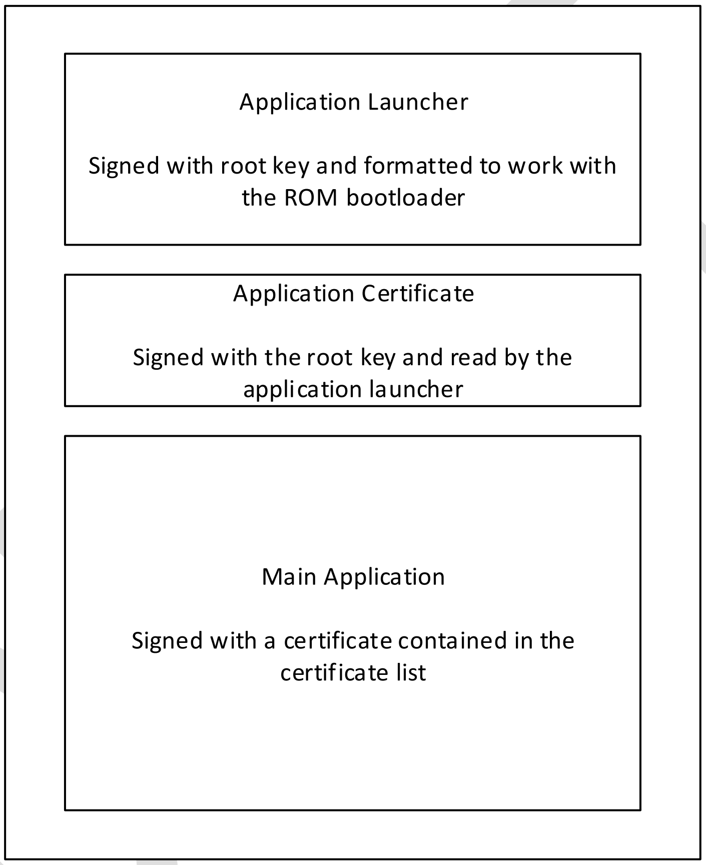

Application Launcher
Signed with root key and formatted to work with the ROM bootloader
Application Certificate
Signed with the root key and read by the application launcher
Main Application
Signed with a certificate contained in the certificate list

Figure 5 Firmware Image Format

3.1 Boot-Time Verification

While booting, the verification chain starts with the ROM bootloader verifying and running the application launcher. The application loader verifies the application certificate, then uses this certificate to verify the main application. If the main application is determined to be valid, it is executed.

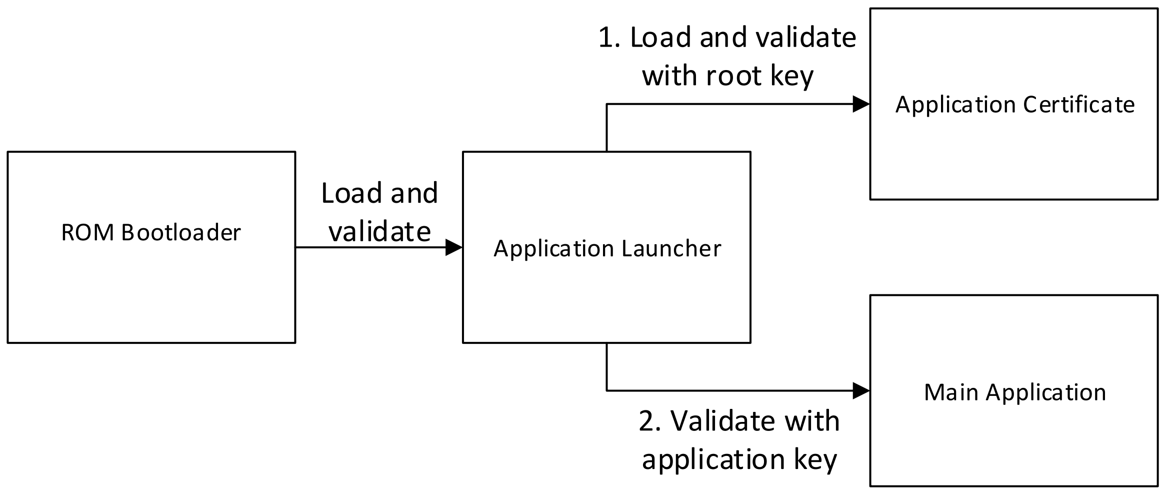


Figure 6 Boot-Time Verification Flow

3.2 Image Update Verification

When verifying an image for a firmware update, the main application must be able to verify all pieces of the new image. The main application will parse each component of the firmware image to ensure that it is valid prior to starting the update.

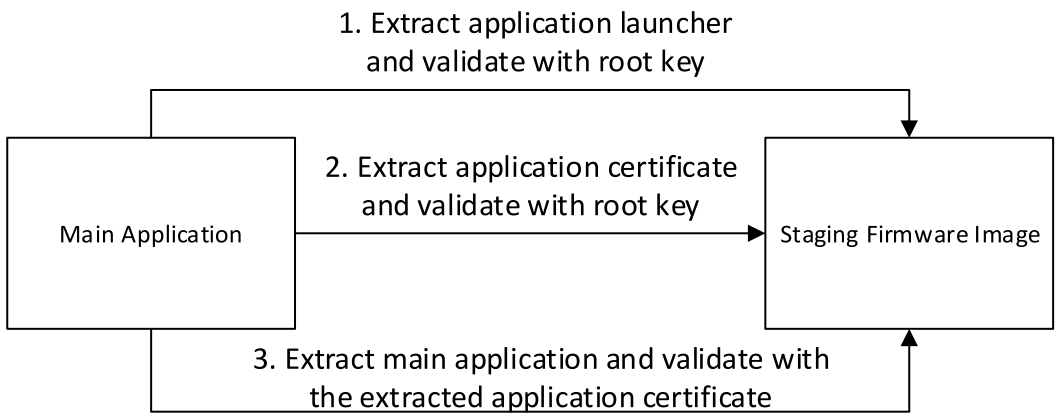

1. Extract application launcher and validate with root key
Main Application
2. Extract application certificate and validate with root key
Staging Firmware Image
3. Extract main application and validate with the extracted application certificate

Figure 7 Update Verification Flow

3.3 Application Launcher Format

The image format of the application launcher is determined by the ROM bootloader of the RoT.

3.4 Application Certificate Format

The application certificate will immediately follow the application launcher in the firmware image, and contains the public keys for the certificates used for component validation. It also indicates which application certificates have been revoked. Note: The RSA 2048 can be changed to RSA 4096 or ECC Key depending on operation mode.

Length (Bytes)	Value
1	Certificate ID A certificate ID shall have at most one bit set. A certificate ID of 0 can never be revoked and should only be used when the maximum number of revocations has been exhausted.
1	Revocation Bit Mask Each bit represents a certificate ID that has been revoked and should no longer be considered valid.
256	RSA-2048 Application Public Key
256	RSA-2048 PFM Public Key
256	RSA-2048 Root Public Key
256	Certificate Signature This is an RSA-2048 encrypted SHA-256 hash of all previous information using the root key.

Table 1 Application Certificate Format

3.5 Main Application Format

The main application will immediately follow the application certificate in the firmware image. The image contains a header that provides the signature of the application.

Length (Bytes)	Value
4	Application Length
n	Application Data
256	Signature This is an RSA-2048 encrypted SHA-256 hash of all previous information using the application key from the certificate.

Table 2 Main Application Format

3.6 Update Verification

The following diagrams describe the process executed by the RoT to validate a firmware image prior to update.

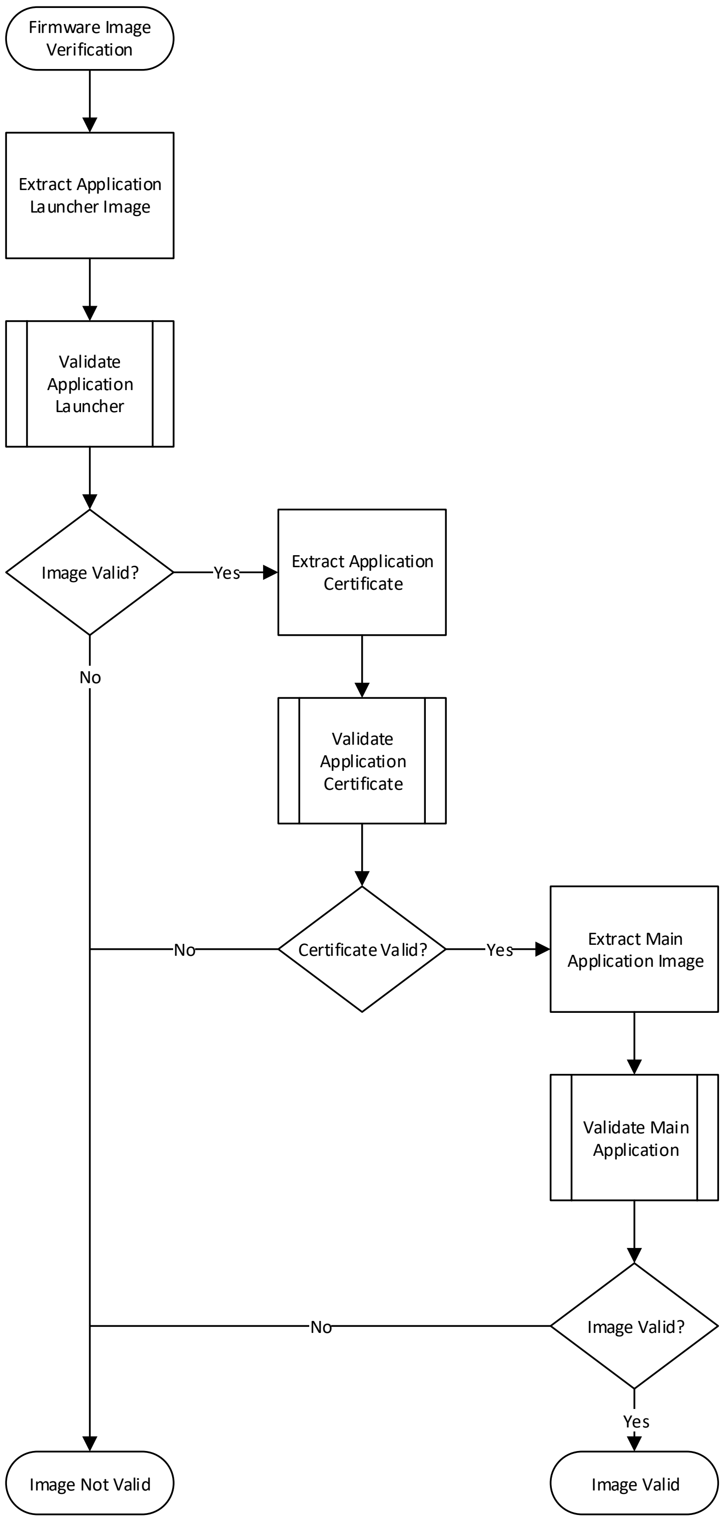
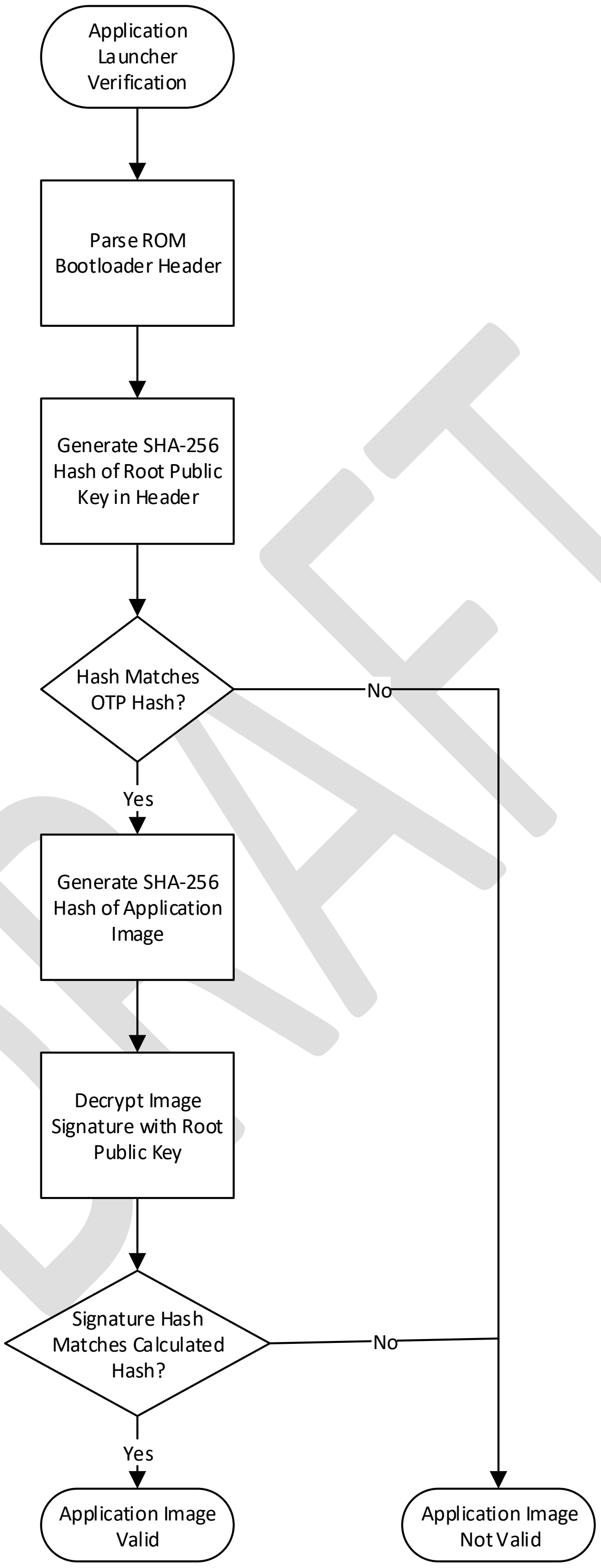

Firmware Image Verification
Extract Application Launcher Image
Validate Application Launcher
Image Valid?
Yes
No
Extract Application Certificate
Validate Application Certificate
Certificate Valid?
No
Yes
Extract Main Application Image
Validate Main Application
Image Valid?
No
Yes
Image Not Valid
Image Valid

Figure 8 Firmware Image Verification



Application Launcher Verification
Parse ROM Bootloader Header
Generate SHA-256 Hash of Root Public Key in Header
Hash Matches OTP Hash?
No
Yes
Generate SHA-256 Hash of Application Image
Decrypt Image Signature with Root Public Key
Signature Hash Matches Calculated Hash?
No
Yes
Application Image Valid
Application Image Not Valid

Figure 9 Application Launder Verification

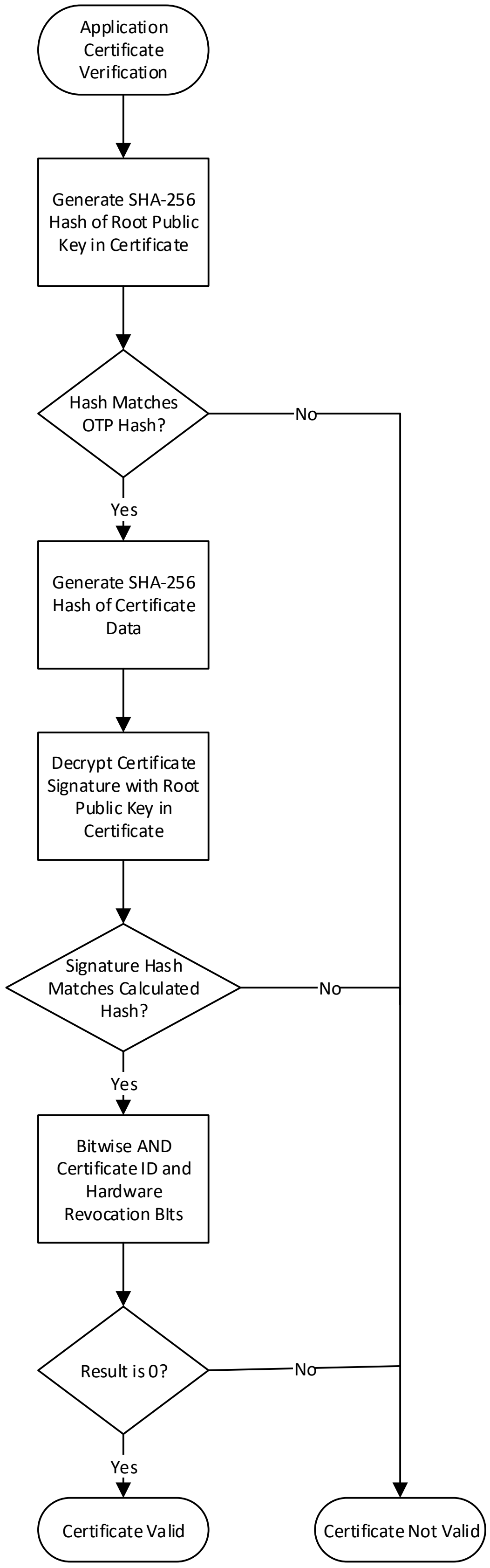

Application Certificate Verification
Generate SHA-256 Hash of Root Public Key in Certificate
Hash Matches OTP Hash?
No
Yes
Generate SHA-256 Hash of Certificate Data
Decrypt Certificate Signature with Root Public Key in Certificate
Signature Hash Matches Calculated Hash?
No
Yes
Bitwise AND Certificate ID and Hardware Revocation BIts
Result is 0?
No
Yes
Certificate Valid
Certificate Not Valid

Figure 10 Application Certificate Verification

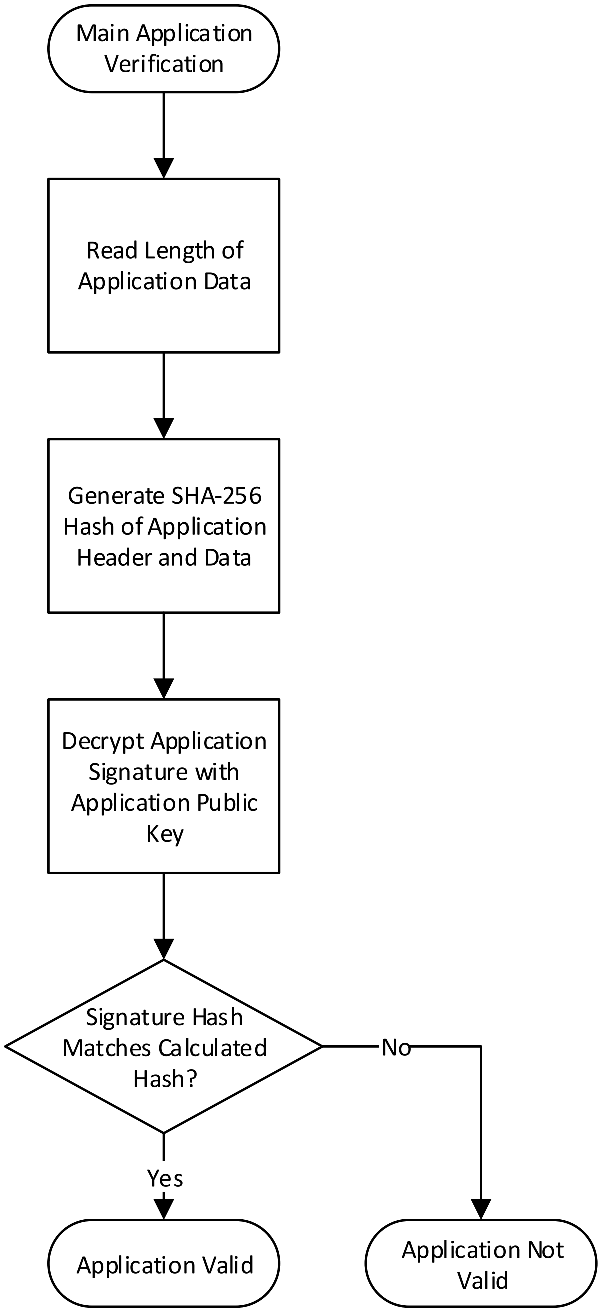

Main Application Verification
Read Length of Application Data
Generate SHA-256 Hash of Application Header and Data
Decrypt Application Signature with Application Public Key
Signature Hash Matches Calculated Hash?
No
Yes
Application Valid
Application Not Valid

Figure 11 Main Application Verification

4 Platform Firmware Manifest (PFM) Updates

The Cerberus RoT firmware accesses a manifest detailing the firmware allowed for the other devices in the platform. As new firmware becomes available for these components, the Platform Firmware Manifest (PFM) needs to be updated for that component, but it is not desirable to upgrade the entire Cerberus firmware to achieve this update. It is therefore possible to run an update process that is only intended to update PFMs.

4.1 PFM Management

Each component that is being protected by Cerberus has a PFM containing information necessary to verify the authenticity of the firmware for that component. For each PFM, Cerberus manages up to two PFM instances: the active PFM and pending PFM. The active PFM contains the information that is currently being used by Cerberus to validate component firmware. The pending PFM contains updated information that should be used by Cerberus but has not yet been made active.

Prior to initial provisioning, Cerberus will have no PFMs. Once a PFM has been provided to Cerberus and activated, the PFM cannot be removed to return to the unprovisioned state.

4.1.1 PFM Activation

Cerberus will not allow a new PFM to be activated until it has successfully validated flash contents. By doing so, it proves the contents of flash contain an image that is supported by the new PFM and that on the next reboot, Cerberus would allow the component to load the firmware. If this requirement were not in place, it would be easy to get into a situation where an updated PFM removes support for the active image, which would result in an authentication failure on the next reboot. If this failed component were imperative to being able to update PFMs, such as the BMC, the system would be in an unrecoverable state.

The same flow is true for initial provisioning as for PFM updating. The first PFM that is provided to Cerberus for a component will not be activated until it has successfully validated the flash for that component. The flash must be fully validated, in the same way that a component firmware update must be validated, before the first PFM can be activated.

4.1.2 Validation Sequence

With two possible PFMs and two sets of flash, there are four combinations of PFM and flash that may need validation when errors are encountered. Of these combinations, activating a new PFM takes priority over activating a new flash image. This is to ensure that the latest information about which firmware is allowed to run is being used.

The following is the order of validation attempts. The sequence ends at the first successful validation, and any validation combination that is not valid for the current state will be skipped.

1) Pending PFM validates staging flash.
2) Pending PFM validates active flash.
3) Active PFM validates staging flash.
4) Active PFM validates active flash.

4.2 Send PFM Update

The process for sending a new PFM to the Cerberus RoT is similar to the process of sending a new firmware image. The area that will hold the updated PFM must first be erased. This is followed by a sequence of commands that send the PFM data. This process will discard any pending PFM that may have already been present in the device.

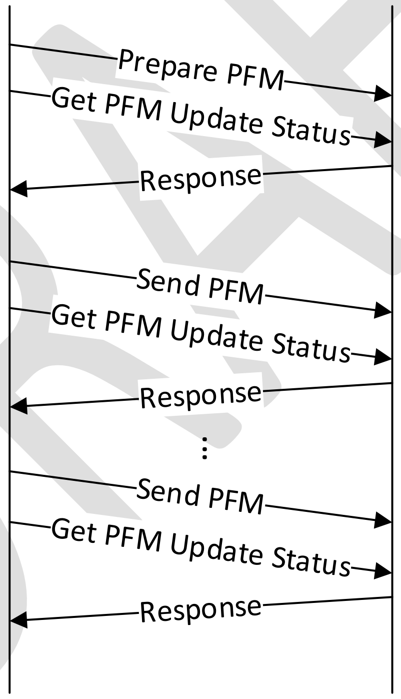


Figure 12 Send New PFM Command Flow

4.3 Update Active PFM

After the new PFM has been transmitted to the RoT, Cerberus can be directed to apply that PFM as the active PFM to use for component validation. Cerberus will first validate that the received PFM is valid. Once the PFM is determined to be valid, it will be marked as the pending PFM. Upon next reboot of the component or of Cerberus, the RoT will attempt to activate the PFM. The update status can be queried to determine if Cerberus accepted or rejected the new PFM.

A valid PFM requires two things:

1) A valid signature using the PFM key from the Application Certificate.
2) An identifier greater than the identifier for the current active PFM.

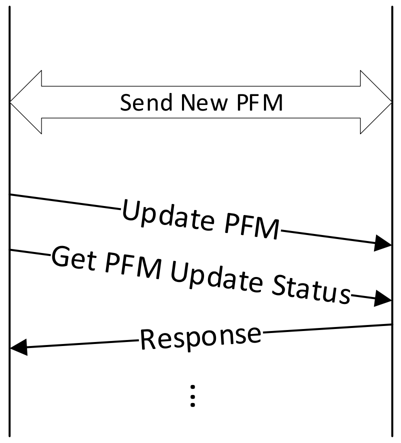
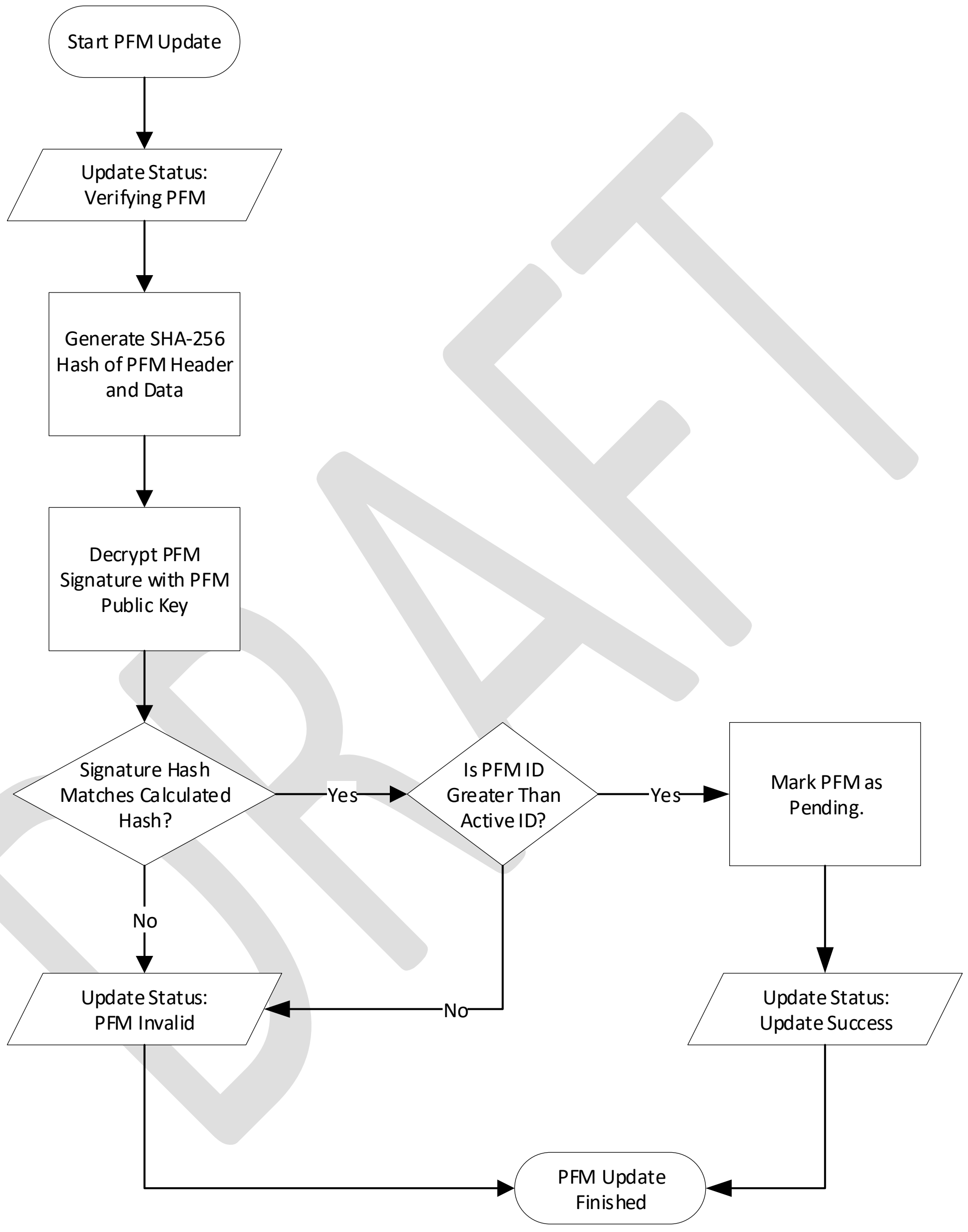


Figure 13 Update PFM Command Flow

4.4 PFM Update Process

The follow diagram describes the process taken by the Cerberus RoT to update the PFM.



Start PFM Update
Update Status: Verifying PFM
Generate SHA-256 Hash of PFM Header and Data
Decrypt PFM Signature with PFM Public Key
Signature Hash Matches Calculated Hash?
Yes
Is PFM ID Greater Than Active ID?
Yes
Mark PFM as Pending.
No
Update Status: PFM Invalid
No
Update Status: Update Success
PFM Update Finished

Figure 14 PFM Update Process