User's Guide
SWRU547A–September 2018–Revised March 2019

Vendor Device Authentication With SimpleLink™ WiFi® Devices

The SimpleLink™ Wi-Fi® Internet-on-a chip™ family of devices from Texas Instruments™ provides a suite of integrated protocols for Wi-Fi® and internet connectivity to simplify the implementation of internet-enabled devices and applications.

As part of the multi-layer security approach of the SimpleLink™ Wi-Fi®, the CC32XXS/SF devices maintain a secure boot sequence, file system security features, and secure sockets. These security features perform authentication to verify that a signature received from trusted chain. TI™ publishes a common trusted certificate list, which is used to authenticate the trusted chain. Vendors that are required to use their own trusted chain should use their own certificate catalog.

This document describes how to authenticate an application that is signed by vendor's certificate using his own certificate catalog, how to store and use it, and how to update it over the air if needed.

Contents

List of Figures

List of Tables

Trademarks

SimpleLink, Internet-on-a chip, Texas Instruments, TI are trademarks of Texas Instruments.
Wi-Fi is a registered trademark of Wi-Fi Alliance.

1 Terms and Abbreviations

Table 1. Terms and Abbreviations

Abbreviation / Term	Description
Authentication	A process that establishes the origin of information or determines an entity's identity.
Boot loader	Software that typically starts after the MCU exits from reset, and is responsible for managing the boot sequence and for loading the target MCU application.
Certificate Authority (CA)	A trusted entity that issues certificates used to verify identities.
Certificate Catalog	The root certificate catalog holds the entire trusted root certificate known to the SimpleLink CC31xx/CC32xx device
Certificates	Certificates are standard-formatted files. They typically contain the public key of the subject and signature of the data.
Digest	The output of a cryptographic hash function.
Integrity	Attribute describing an object that remains intact, in its entirety, compared to its original version.
OTP	One-time-programming. A memory block that could be programmed only once.
PKI	Public key infrastructure is a technology for authenticating users and devices in the digital world.
Root-of-trust	The initial source of trust in the system.

2 Supported Platforms

The Vendor Certificate Catalog is supported on the following hardware platforms:

- CC32XXS
- CC32XXSF

3 Minimum Requirements

- Uniflash version 4.5 and above
- Service pack version sp_3.9.0.6_2.0.0.0_2.2.0.6 and above for CC3220S/SF
- Service pack version sp_4.1.x.x_3.1.x.x_3.1.x.x and above for CC3235S/SF
- Serial Flash device: Macronix MX25R3235FM1IH0

4 Constraints

Factory defaults and gang programming are not supported while using this feature.

5 Overview

To achieve a secure boot sequence, the system manages the following steps:

1. Authenticate a file cryptography based on PKI during programming and updating of a file.
2. If this step passes, calculate a digest for the file and store it encrypted.
3. Verify the integrity of the file every boot based on the calculated encrypted digest.

The root of the verified chain must be contained in the trusted list of the system. By default, this list holds trusted root certificates that are approved by TI, and the list is signed by TI.

Using the vendor certificate catalog allows the vendor to choose their own certificates and create their own list.

The vendor certificate catalog is verified based on the vendor's certificate, which is stored in one time programming (OTP) memory. This method binds between the hardware to the vendor.

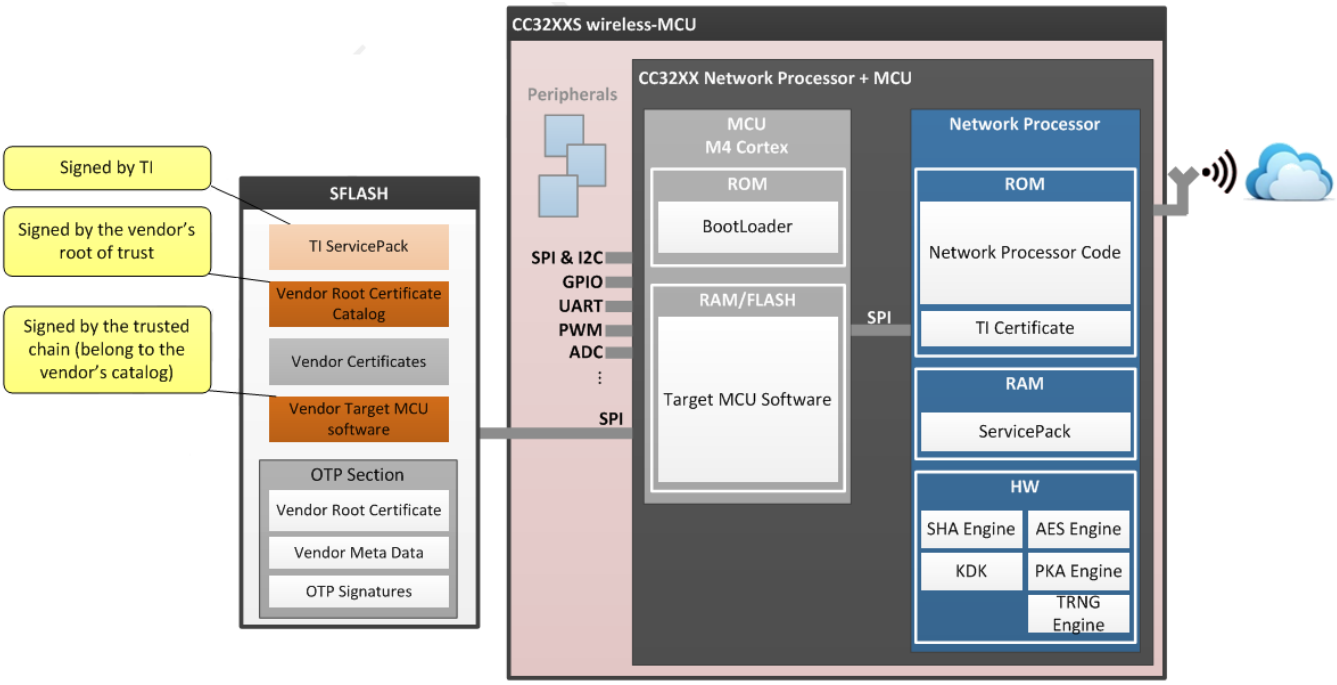
The vendor's certificate is kept on OTP memory, thus it cannot be replaced once programmed. In the OTP section, the vendor can add device-specific information and sign it together, which allows the vendor to verify that the hardware platform is authentic and produced by the vendor.

This feature adds the following capabilities:

- The ability to use a trusted root certificate catalog signed by the vendor's root-of-trust.
- The ability to use the vendor's root-of-trust to verify the target MCU software. The root of trust of the vendor can be based on a self-signed certificate or a certificate signed by a well-known CA. When programmed, the root of trust cannot be replaced or changed.
- The vendor catalog can be updated over-the-air.
- The ability to authenticate that a board or product is manufactured by the vendor.
- The ability to protect the target MCU software and the vendor's root certificate catalog against downgrade during updates.
- The ability to protect the network processor service pack against downgrade during updates.

Figure 1 shows the architecture.

Figure 1. Architecture Scheme



The basic authentication flow is handled entirely by TI: TI supplies the list of trusted root-CAs that a vendor can use, and this list is signed by TI's certificate.

In the basic authentication flow, only applications and files that require authentication must be signed by a certificate with its root-CA as a part the certificate catalog.

The certificate catalog in this flow is signed by TI's certificate, which is stored in the ROM of the network processor. The same certificate is used to validate TI's service pack and TI's files.

Figure 2 shows the basic authentication flow.

In the customized authentication flow, the vendor can sign the application and files with his own certificate and his own certificate catalog.

The vendor certificate catalog is signed by the vendor's certificate, which is stored in the OTP section.

TI's service pack and files are still validated by TI's certificate.

Figure 3 shows the customized authentication flow.

Figure 2. Basic Authentication Flow

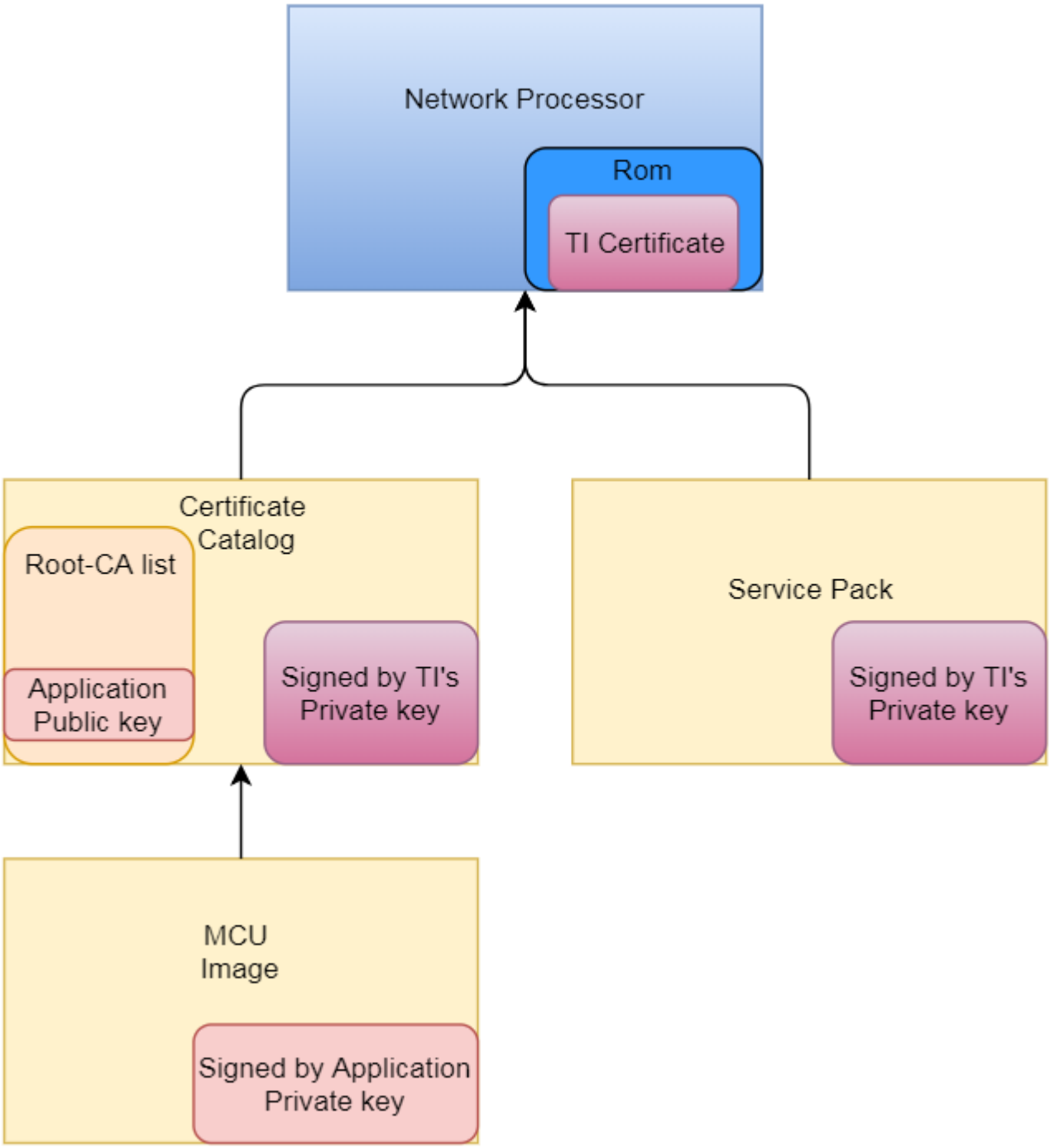
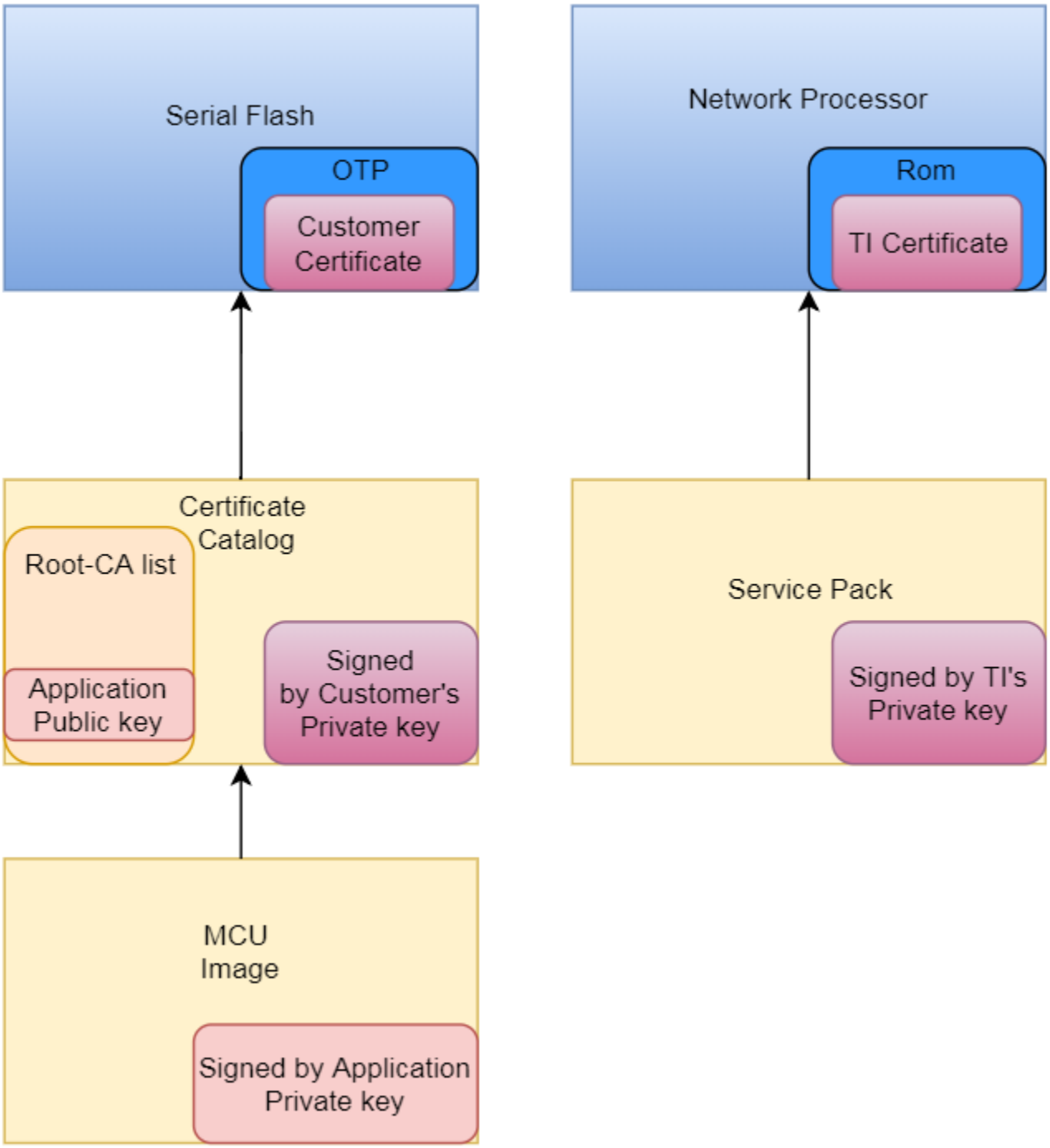


Figure 3. Customized Authentication Flow



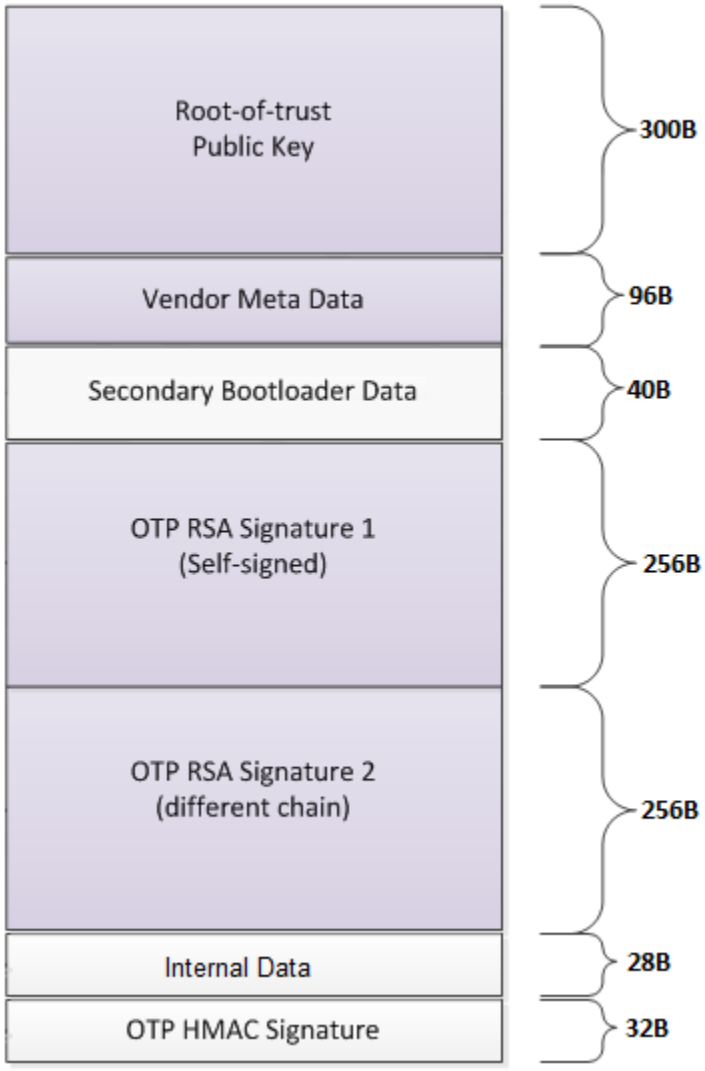
6 OTP Overview

The OTP block contains the root-of-trust of the system and is a fundamental block for the vendor certificate catalog. This block binds the hardware to a specific vendor by allowing a specific hardware to load only applications that are signed by this specific vendor.

The block size is 1008 Bytes, and is programmed automatically by the system during the initial programming. This feature does not need to change the vendor's factory setup, thus no hardware changes are needed, only new image and OTP files.

Figure 4 shows the target content of the OTP block.

Figure 4. OTP Block



The OTP block contains mandatory data and might contain additional data that is optional; when the optional data is not provided, these sections will be padded with “0”.

The following list provides information on this data and whether it is mandatory or optional:

- Root of trust public key [300 Bytes] [Mandatory] – This public key is an RSA key that functions as the vendor’s root-of-trust. The vendor’s certificate catalog must be signed by the corresponding private key. Other secure and signed files should be signed with one of the chains in the catalog or optionally by this key. The public key must be RSA-1024 key or RSA-2048 key.

- Vendor Meta Data [6 Bytes] [Mandatory] – MAC address part of the meta data, MAC address of zeros allows programming the OTP file on all devices or specific MAC for specific devices.
- Vendor Meta Data [90 Bytes] [Optional] – General purpose block of data that the vendor can choose to use in the application.

NOTE: The Vendor Meta Data [90 Bytes] is not supported by the current release.

- Secondary boot loader data [40 Bytes] – Managed by TI.
- OTP RSA Signature 1 [256 Bytes] [Mandatory] – RSA signature (SHA256) of the first 3 sub-blocks that could be verified with the root-of-trust public key. This signature must be provided during the programming.
- OTP RSA Signature 2 [256 Bytes] [Optional] – A signature of the first 3 sub-blocks that could be verified with a hardcoded public key on the target MCU software. This signature allows the target MCU software to authenticate that the hardware was produced by the vendor. The root-of-trust in this authentication process is inside the MCU software itself, and the data verified by default contains device-specific information (optionally, selected by the vendor). When the signature is different from device to device (the data contains device specific information), this signature should be kept on the OTP block and the software could only have the public key. If the vendor decides to remove the device-specific information from the OTP block, the signature is equal to all devices and can be hardcoded in the software itself.

NOTE: The OTP RSA Signature 2 is not supported by the current release.

- Internal data [28 Bytes] [Mandatory] – Managed by TI.
- OTP HMAC Signature [32 Bytes] [Mandatory] – This HMAC signature is calculated and kept by the device. The value is calculated on the first 4. This signature is used to verify the integrity of the OTP block.

The OTP block can be divided into three parts:

- OTP meta data, which contain the root of trust public key, vendor meta data, and secondary boot loader data.
- Signature of the OTP meta data
- Vendor signature

Combining these parts gives the OTP information file, which is programmed on the device.

The vendor must add this block to the image during the image creation. For more information on how to prepare this data, refer to Section 7.2.

7 Getting Started

The following describes the procedure for creating a certificate catalog, building the OTP file, and how to use it.

To create the certificate catalog, the OTP, and program the device, the latest UniFlash must be installed: refer to the *UniFlash CC3120,CC3220 SimpleLink™ Wi-Fi® and IoC™ Solution ImageCreator and Pro User's Guide*.

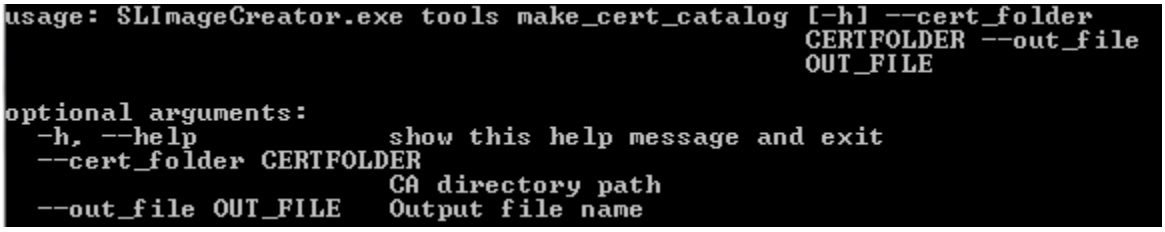
The following steps must be run from the command line window in the Image creator installation folder (the Chapter Command Line in the *UniFlash CC3120,CC3220 SimpleLink™ Wi-Fi® and IoC™ Solution ImageCreator and Pro User's Guide*).

7.1 Certificate Catalog Creation

7.1.1 Creating the Certificate Catalog

1. Copy the root-ca certificates in DER format that must be in the catalog to a specific folder; for example: ExampleKnownCA. This list must contain all SSL root-CAs required by the application and all the root-CAs that are required for the application and files authentication. Limitation: 100 root-ca certificates can be stored in the certificate catalog.
2. Use the command line command which creates the certificate catalog – make_cert_catalog, as shown in Figure 5.

Figure 5. make_cert_catalog Usage



Function Example:

```
SLImageCreator.exe tools make_cert_catalog --cert_folder "C:\Certs\ExampleknownCA" --
out_file "C:\Certs\certificate_Catalog.lst"
```

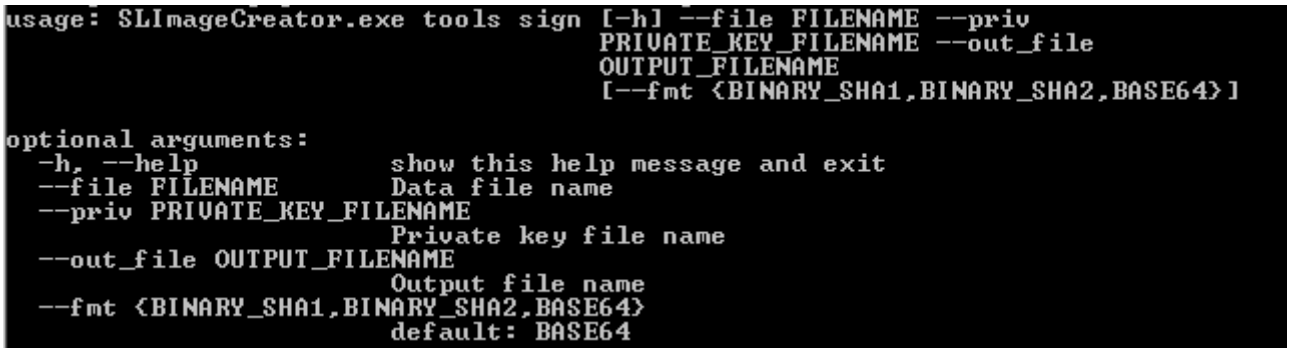
Running this function creates the certificate_Catalog.lst file from all the certificates from the ExampleknownCA folder.

All the root-ca certificates must be in DER format.

7.1.2 Signing the Certificate Catalog

Use the command line command which signs the certificate catalog – sign, as shown in Figure 6.

Figure 6. certificate catalog sign Usage



Function Example:

```
SLImageCreator.exe tools sign --file "C:\Certs\certificate_Catalog.lst" --
priv "C:\Certs\vendor_key.pem" --out_file "C:\Certs\certificate_Catalog.lst.signed.bin" --
fmt "BINARY_SHA1"
```

Running this function signs the certificate_Catalog.lst file in SHA1 format using the vendor_key.

For the CC3220S and CC3220SF, the certificate catalog must be signed in SHA1 format. For the CC3235S and CC3235SF, the certificate catalog must be signed in SHA2 format.

7.2 OTP Creation

To create the OTP section, follow these steps:

1. Create the OTP metadata file.
2. Sign the meta file.
3. Create the OTP information file.

7.2.1 Creating the OTP Metadata File

1. Use the command line command which creates the OTP meta file – meta, as shown in Figure 7.

Figure 7. meta Usage

```
usage: SLImageCreator.exe tools meta [-h] --cert CERT --out_file OUT_FILE
                                     [--mac MAC_ADDR] [--usechain]

optional arguments:
  -h, --help            show this help message and exit
  --cert CERT           Vendor certificate file name
  --out_file OUT_FILE   Output file name
  --mac MAC_ADDR        Mac Address
  --usechain            if signature 2 will be used
```

2. Using MAC address 000000000000 allows using the OTP file on all devices. Using a specific MAC address binds the file to the same MAC address.
3. When creating the inf file, if --usechain is used in the creation of the metafile, the OTP RSA Signature 2 must be entered.

Function Example:

```
SLImageCreator.exe tools meta --cert "C:\Certs\vendor_root-ca.pem" --
out_file "C:\OTP\vendor_otp.meta" --mac "112233445566" --usechain
```

Running this function creates a vendor_otp.meta file with the public key of vendor_root-ca.pem. The MAC address of the device is 112233445566, and it will demand both RSA signatures.

7.2.2 Signing the OTP Metadata File

1. Use the command line command which creates the OTP meta file – sign, as shown in Figure 8.

Figure 8. sign Usage

```
usage: SLImageCreator.exe tools sign [-h] --file FILENAME --priv
                                     PRIVATE_KEY_FILENAME --out_file
                                     OUTPUT_FILENAME
                                     [--fmt {BINARY_SHA1,BINARY_SHA2,BASE64}]

optional arguments:
  -h, --help            show this help message and exit
  --file FILENAME       Data file name
  --priv PRIVATE_KEY_FILENAME
                        Private key file name
  --out_file OUTPUT_FILENAME
                        Output file name
  --fmt {BINARY_SHA1,BINARY_SHA2,BASE64}
                        default: BASE64
```

2. The meta file can be signed in SHA1 or SHA2, using --fmt BINARY_SHA1 or --fmt BINARY_SHA2.

Function Example:

```
SLImageCreator.exe tools sign --file "C:\OTP\vendor_otp.meta" --priv "C:\Certs\vendor_key.pem" --
out_file "C:\OTP\vendor_otp.meta.sig" --fmt "BINARY_SHA2"
```

Running this function signs the vendor_otp.meta file in SHA2 format using the vendor_key.

7.2.3 Create the OTP Information File

1. Use the command line command which creates the OTP meta file – inf, as shown in Figure 9.

Figure 9. inf Usage

2. The meta file signatures are signed in SHA1 or SHA2; the inf function must know which format has been used by using --algo 1 for BINARY_SHA1 or --algo 2 for BINARY_SHA2.

Function Example:

```
SLImageCreator.exe tools inf --algo 2 --sign1 "C:\OTP\vendor_otp.meta.sig" --
sign2 "C:\OTP\vendor_otp.meta.sig" --meta "C:\OTP\vendor_otp.meta" --
out_file "C:\OTP\vendor_otp.inf"
```

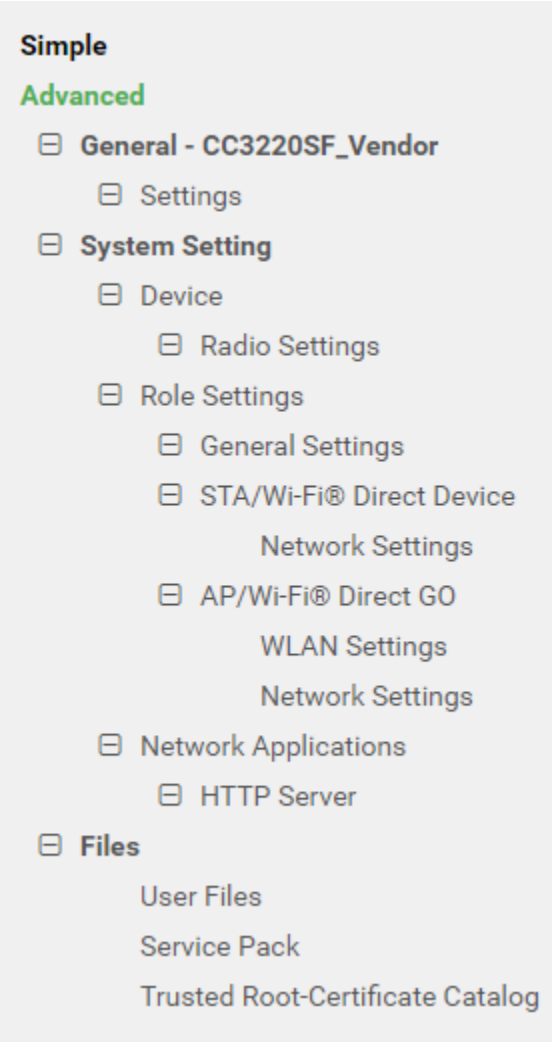
Running this function creates the vendor_otp.inf file using both RSA signatures that were created using SHA2 format.

The vendor_otp.inf is the file that is programed into the OTP section of the device.

7.3 OTP Programming Using Image Creator

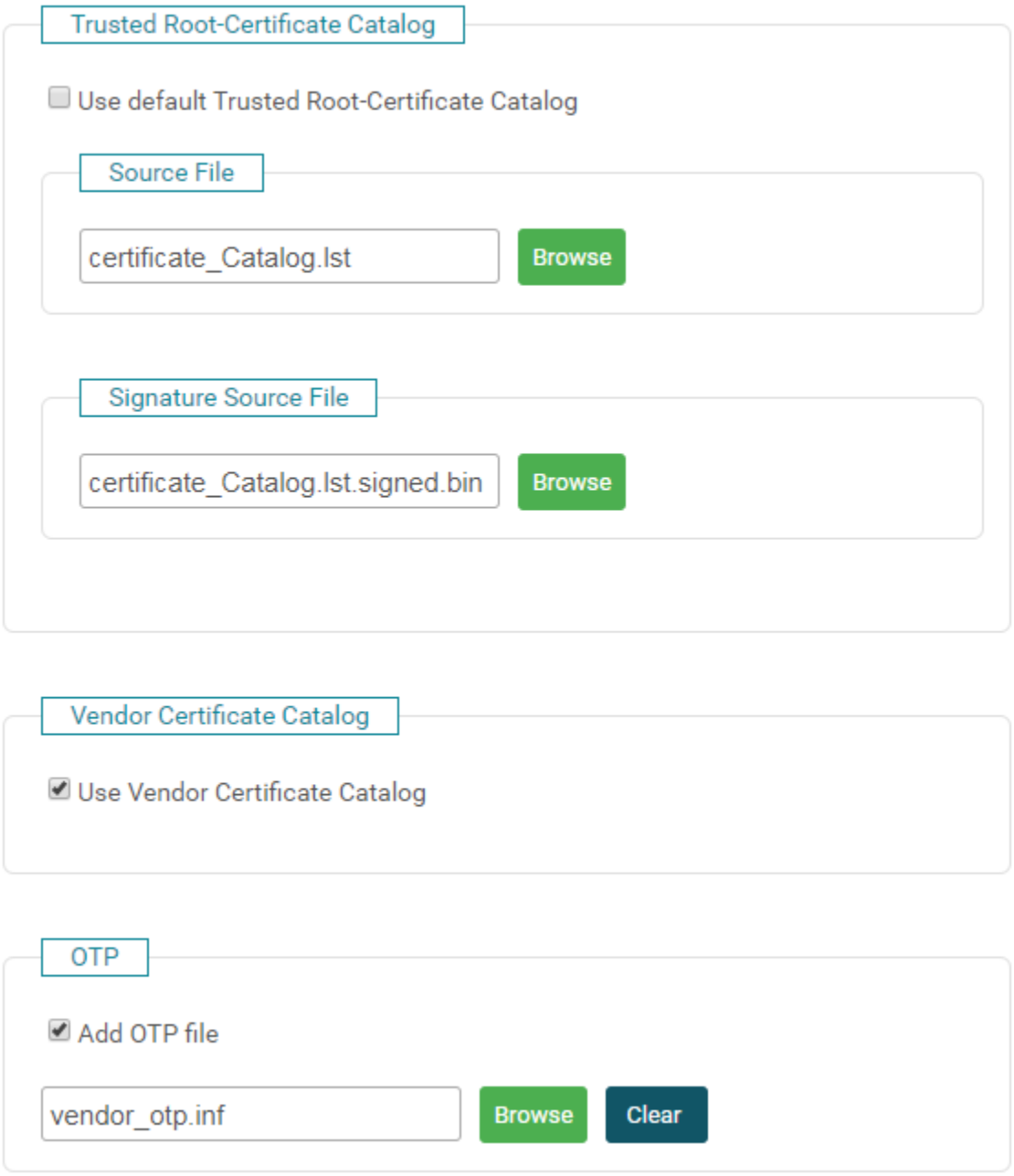
1. Create a new project, or open an existing project, in the Image Creator. Add all the required OTA files, including the service pack. See the Use, Creating a New Project, and Opening a Recent Project chapters in the *UniFlash CC3120,CC3220 SimpleLink™ Wi-Fi® and IoC™ Solution ImageCreator and Pro User's Guide*.
2. Change the view to Advanced, as shown in Figure 10.

Figure 10. Advanced View



3. In the Trusted Root-Certificate Catalog, change the catalog files to the Certificate_Catalog.lst and certificate_Catalog.lst.signed.bin created in Section 7.1.1.
4. Enable the Use Vendor Certificate Catalog option.
5. Enable the Add OTP file and choose the vendor_otp.inf created in Section 7.2.3, as shown in Figure 11.

Figure 11. Catalog View



7.4 OTP OTA Using Image Creator

1. Create a new project, or open an existing project, in the Image Creator. Add all the required OTA files, including the service pack. Refer to the Use and Create an OTA chapters in the *UniFlash CC3120,CC3220 SimpleLink™ Wi-Fi® and IoC™ Solution ImageCreator and Pro User's Guide*.
2. Change the view to Advanced.

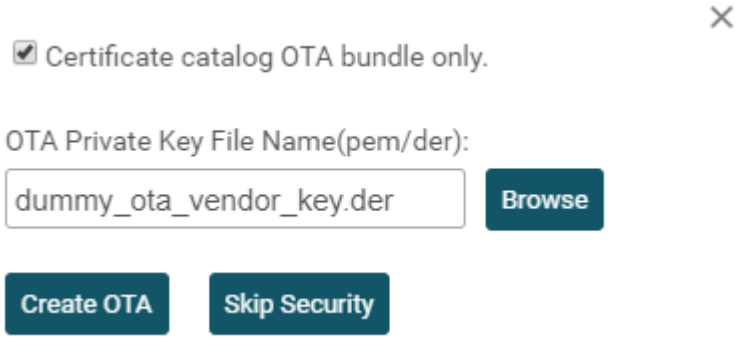
When the OTP is programmed, downgrading the certificate catalog or service pack is not allowed.

The image in the OTA bundle must be signed by the private key of the OTP public key.

7.4.1 Create OTA Bundle That Holds the Certificate Catalog Only

1. Choose the Burn option, then Create OTA.
2. Check the Certificate catalog OTA bundle only option.
3. Choose the Private key needed for the OTA, and press Create OTA, as shown in Figure 12.

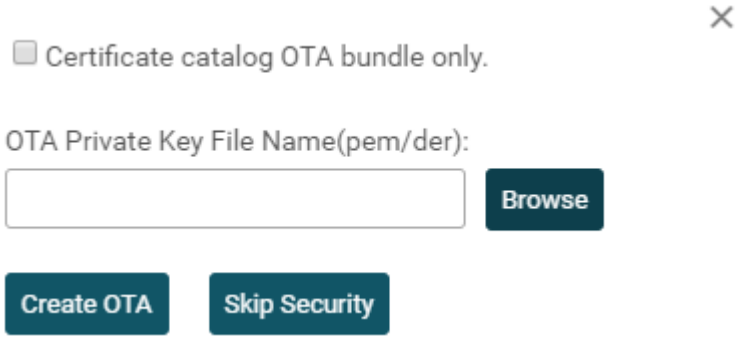
Figure 12. Certificate Catalog OTA



7.4.2 Create OTA Bundle

1. Choose the Burn option, then Create OTA.
2. Check that the Certificate catalog OTA bundle only is unchecked.
3. Choose the Private key needed for the OTA, and press Create OTA, as shown in Figure 13.

Figure 13. OTP OTA Bundle



Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (September 2018) to A Revision	Page
• Updated Service Packs in Minimum Requirements section.	2
• Added Note.	7
• Updated Signing the Certificate Catalog section.	8

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265