Application Note

Keywriter Deployment Strategies: Secure Device Provisioning in Production Environments

Diwakar Dhyani,Frangline Jose, Keerthy J

ABSTRACT

This application note provides a comprehensive guide for converting TDA4x/DRA8x processors from High Security Field-Securable (HS-FS) mode of operation to High Security Security-Enforced (HS-SE) mode of operation. The document explains TI's One Time Programmable (OTP) Keywriter tool architecture and deployment methods for development and production environments. The document also explains field update procedures by using Texas Instruments Foundational Security (TIFS) API. Engineers will learn how to select appropriate secure key provisioning strategies, implement secure device programming workflows, and troubleshoot common issues encountered during OTP key programming operations.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

Copyright © 2026 Texas Instruments Incorporated

1 Introduction

This document is designed for production engineers, security architects, and manufacturing teams implementing secure key provisioning for TI processors. It covers Keywriter deployment strategies, architecture components, and production workflows. Keywriter users will gain practical knowledge to select appropriate deployment methods, transition the device from field-securable (FS or HSFS) to security-enforced (HS-SE) state, and resolve common provisioning issues. Readers will gain both theoretical understanding and practical knowledge needed to implement, optimize, and troubleshoot Keywriter operations in real-world production environments.

1.1 Why Keywriter is Needed

TI processors ship from the factory in HS-FS (High Security Field-Securable) mode. In this state, secure-boot is not enforced since root-of-trust (RoT) keys are not yet programmed into the device OTP-eFuses. The Keywriter tool enables us to program these root-of-trust keys and the associated security fields into the eFuse, transitioning devices from HS-FS to HS-SE mode of operation.Secure boot is enforced by hardware when the device is in HS-SE state.

The following fields can be programmed in the OTP-eFuse array, but not all the fields are mandatory to transition to HS-SE device:

Mandatory Fields

- SMPK HASH (Secondary Manufacturer Public Key Hash).
- SMEK (Secondary Manufacturer Encryption Key).
- KEYCNT (Key Count).
- KEYREV (Key Revision).

Optional Fields

- BMPK HASH (Backup Manufacturer Public Key Hash).
- BMEK (Backup Manufacturer Encryption Key).
- SWREV, MSV, JTAG (Software Revision for anti-rollback protection, Model Specific Value, JTAG disable).
- Extended OTP fields.

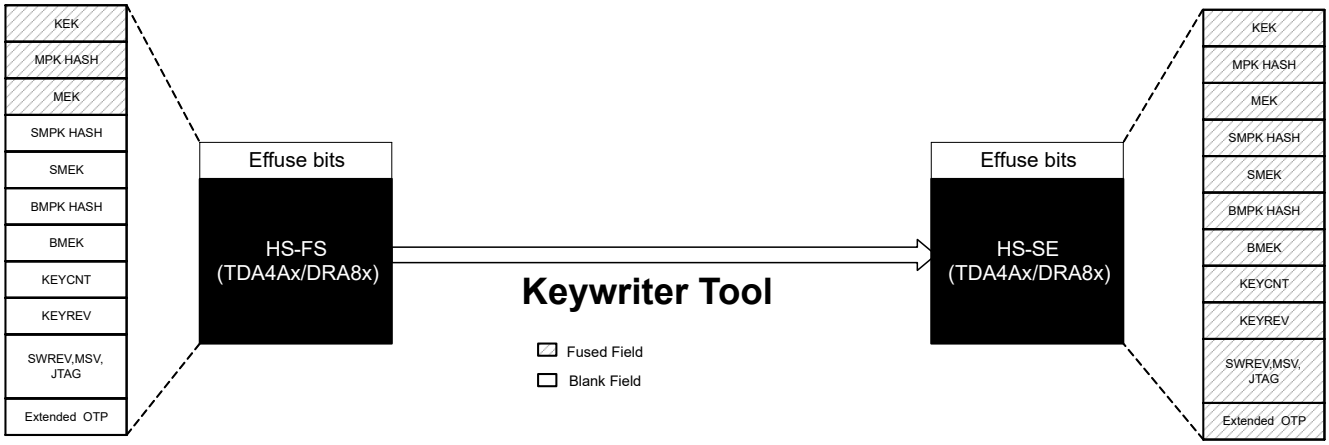

Figure 1-1. Device Lifecycle: HS-FS to HS-SE Transition via Keywriter eFuse Programming Workflow

Copyright © 2026 Texas Instruments Incorporated

1.2 What is Keywriter

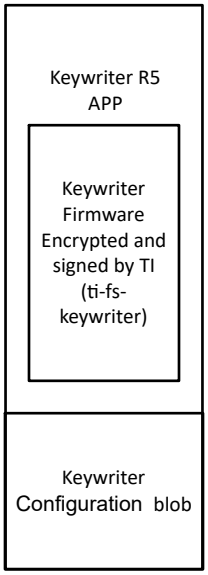

Figure 1-2. Keywriter System Architecture Showing TIFS Firmware, R5 Application, and Certificate Blob Components

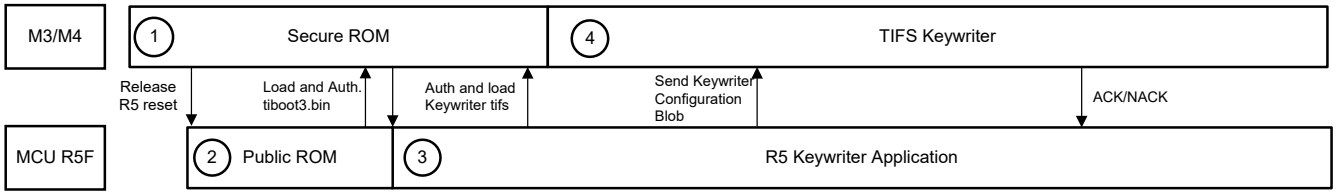

Figure 1-3. Keywriter Boot Flow

The Keywriter tool consists of three main components:

- **Keywriter firmware**: This is the secure OTP keywriter firmware that runs on the secure subsystem of the device (M3/M4_0). It is responsible for verifying the OTP configuration from x509 certificate and programming the data into the device OTP-eFuses.
- **Keywriter Configuration Blob**: Certificate containing effuse configuration that gets programmed in the OTP-eFuses.
- **Keywriter R5 APP**: This is the public keywriter application. It boots directly on the primary boot core of the device without any other bootloader. It is responsible for sending the OTP configuration to the secure keywriter component running on the security subsystem (M3/M4) to program the keys into the device OTP-eFuses.

Copyright © 2026 Texas Instruments Incorporated

2 Keywriter Certificate Blob

2.1 Certificate Generation Overview

The Keywriter certificate blob uses a layered security approach combining digital signatures and encryption. The OTP configuration data is encrypted with a randomly generated AES-256 session key. This session key is then encrypted using the TI-FEK (TI Field-Encryption Key), verifying only authentic TI firmware with the corresponding private key can decrypt it. The entire certificate is signed using SMPK (Secondary Manufacturer Private Key) for authentication. Optionally, the certificate can be dual-signed by also using BMPK (Backup Manufacturer Private Key) to enable key rotation scenarios. For complete keywriter blob generation flow refer Figure below.

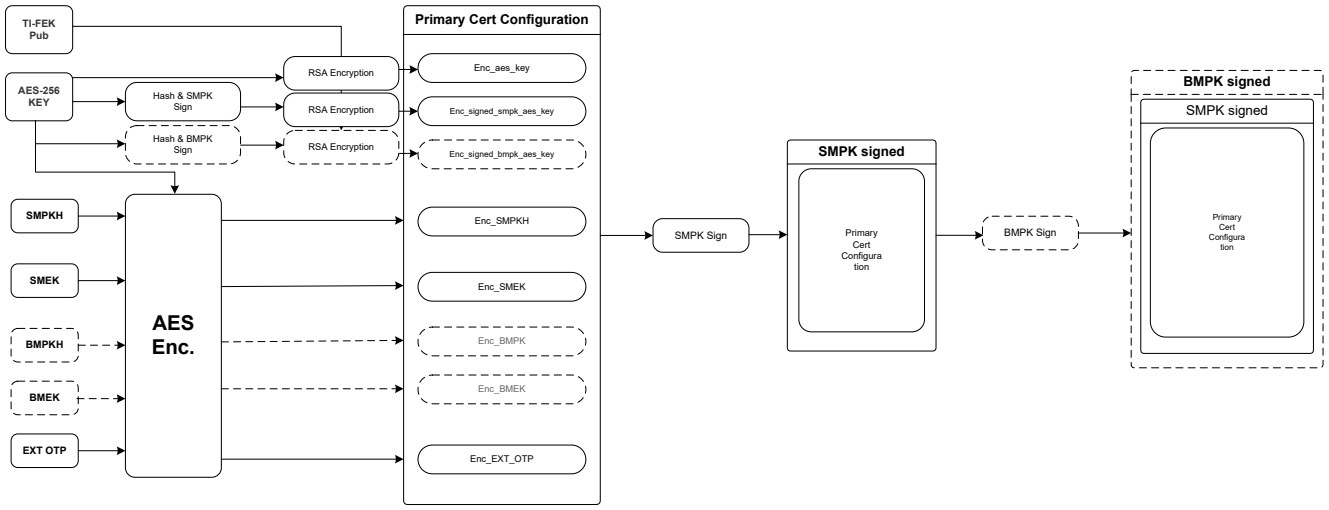

Figure 2-1. Certificate Blob Generation Flow: AES Encryption, TI-FEK Key Wrapping, and Dual-Signature Authentication

Copyright © 2026 Texas Instruments Incorporated

3 Production Deployment Methods

Multiple deployment methods are available to suit different production environments and requirements. Each method is provided with specific implementation steps, allowing customers to select the approach that best fits their production setup and needs. This flexibility ensures that customers can choose any suitable method without being constrained to a predefined use case.

3.1 OSPI Boot Method

Initial Setup and Key Provisioning Process:

1. Flash the Keywriter image to the primary boot location of OSPI flash memory.
2. Flash the HS-SE signed application images to the backup boot location.
3. Power on the processor (MPU). The boot ROM reads the Keywriter image from the primary location in OSPI flash.
4. Since the device is in HS-FS (High Security - Field Securable) mode, the Keywriter application runs and programs the security keys onto the device.

   Transition to Secure Boot:

   Once OTP-eFuses are programmed, the device enters HS-SE mode on the next power cycle. When the customer keys are programmed and KEYREV, KEYCNT is also programmed, the hardware hereafter switches to HS-SE state. This process cannot be reversed back to HS-FS state.
5. On next power-on reset, the ROM bootloader enforces secure boot. The Keywriter image at the primary boot location is loaded first, but it fails authentication because it is not signed with the customer root-of-trust keys newly programmed in OTP-eFuses.
6. The ROM bootloader's fallback mechanism activates, reading the application image from the backup boot location of OSPI Flash. It successfully authenticates the HS-SE signed application image using the eFuse-stored public key hash.
7. The device successfully boots using the image that is signed with customer keys (SMPKH).
8. At this stage it is good practice to erase the primary image in the OSPI Flash and re-program it with a copy of the application image signed with customer keys that is the same as the one loaded from the secondary partition of the OSPI Flash.

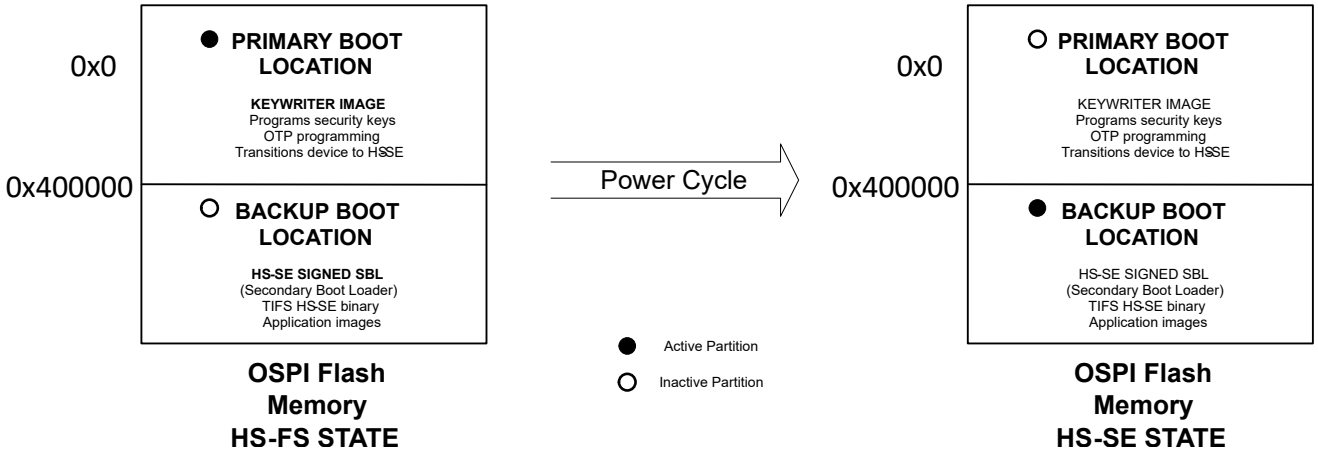

Figure 3-1. OSPI Boot Method

Copyright © 2026 Texas Instruments Incorporated

3.2 eMMC Boot Method

Initial Setup and Key Provisioning Process:

1. Flash the Keywriter image to the Boot 0 partition of the eMMC device.
2. Flash the HS-SE signed application images to the Boot-1 partition of the eMMC device.
3. Configure the boot mode pins for eMMC boot mode and power on the device.
4. ROM bootloader reads the PARTITION_CONFIG register (default: Boot-0 active), since the device is in HS-FS state, ROM authenticates the Keywriter, and runs it.
5. After the Keywriter completes key provisioning, keywriter application updates the PARTITION_CONFIG field in the eMMC's EXT_CSD register to change the active boot partition from Boot-0 to Boot-1.For reference implementation refer FAQ.

Transition to Secure Boot:

6. Power cycle the board, the ROM bootloader checks the boot mode pins (configured for eMMC boot mode) and reads the PARTITION_CONFIG field to determine which partition is active.
7. Since the active partition was changed to Boot-1 in the previous step, the ROM fetches the image from Boot-1.
8. The device successfully boots using the HS-SE signed production images from Boot-1
9. At this stage it is good practice to erase the primary image in the eMMC Flash and re-program it with a copy of the application image signed with customer keys that is the same as the one loaded from the secondary partition of the eMMC Flash.

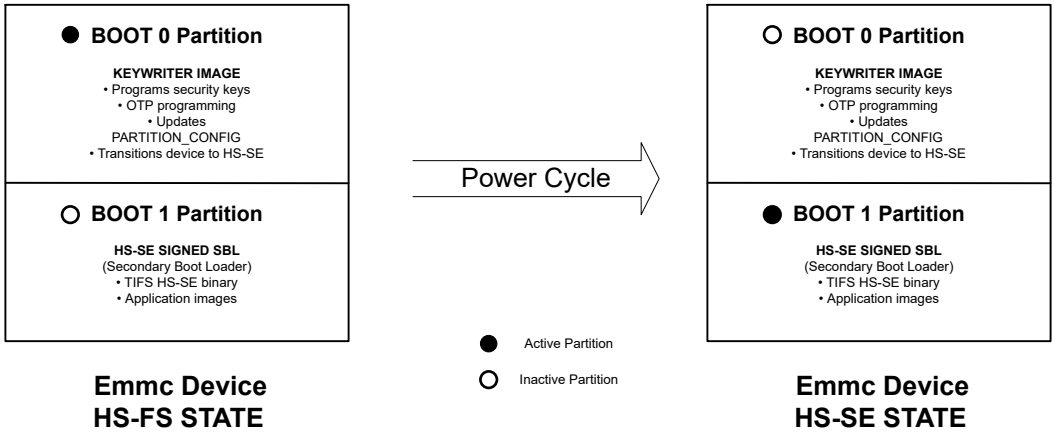

Figure 3-2. eMMC Boot Mode

Copyright © 2026 Texas Instruments Incorporated

3.3 eMMC Boot with DFU Backup Method

Initial Setup and Key Provisioning Process:

1. Set the Primary boot media as EMMC and backup as DFU.
2. Power on the device. As we haven't flashed any image into the eMMC, boot ROM will not be able to read the image from primary boot media and falls back to backup boot mode.
3. Boot ROM initiates DFU boot. Load the keywriter image using DFU interface.
4. The keywriter binary programs the keys to OTP.

Transition to Secure Boot:

5. Power cycle the board, Boot till U-Boot using HS-SE signed image via DFU interface.Flash the HS-SE signed images to emmc using dfu interface as mentioned in section "5 eMMC flashing" of TDA4 Flashing Techniques.
6. Power cycle the board, since emmc is the primary boot media and has valid signed boot image. ROM will read, authenticate and boots the application image from emmc.

Note

Backup boot location is different than the Backup boot mode. The Boot ROM supports a backup OSPI offset if it doesn't find a valid image at the primary offset ,and it automatically jumps to a backup offset within the same OSPI flash. Whereas Backup boot mode is a hardware-configured fallback to a completely different boot media It is determined by physical BOOTMODE pins.

3.4 JTAG/Development Boot Method

Initial Setup:

1. Configure the device to Development boot (DEV BOOT) mode by setting the appropriate boot pins (BOOTMODE[4] = 0).
2. During DEV BOOT, the device waits for a firmware loaded message from MCU R5.
3. Connect JTAG emulator (XDS110, XDS200, or XDS560) to microprocessor/MPU.

Provisioning Process:

4. Connects to MCU Cortex R5F Core 0. Resets the R5F core to clean state.
5. Load keywriter application onto the MCU R5 core.
6. Load keywriter certificate blob to designated memory location (e.g., 0x41C7F004 for TDA4AL,KeyWriter certificate starts from keywr_end + 1, which is placed in section ".keywr_bin_end" Please look up map file or the linker command file).
7. Initiate execution of keywriter program on MCU R5.
8. Keywriter firmware validates certificate, decrypts keys, and programs OTP-eFuses, transitioning device from HS-FS to HS-SE.

The FAQ in this link has more details about this method. This FAQ also provides a java script that can directly be used with CCS scripting shell.

3.5 RGMI Ethernet Boot Mode

Initial Setup and Key Provisioning Process:

1. Configure the device in RGMI boot mode.
2. On the host PC set the Static IP for the microprocessor and configure the link to 100M full duplex.
3. On the host PC download the tftpd32 application and set up the DHCP server using the tftpd32 application.
4. Connect the PC ethernet port to the CPSW port on the device using an ethernet cable.

For more details, see the FAQ.

Note

Ethernet boot mode is only supported on TDA4VH class of devices

Copyright © 2026 Texas Instruments Incorporated

4 Updates in the Field

4.1 SWREV (Software Revision)

The SWREV field in OTP-eFuses is used to implement software rollback protection, preventing attackers from downgrading the primary application firmware to older versions with known vulnerabilities. When a device boots, TIFS/BOOT ROM code compares the software revision value embedded in the boot image certificate against the SWREV value programmed into the OTP-eFuses. If the certificate's software revision is less than the OTP SWREV value, the boot process is rejected, ensuring authentication and loading of equal or higher SWREV revision numbers.

Components for which SWREV can be updated are:

1. Software Revision SBL.
2. Software Revision SYSFW/TIFS.
3. Software Revision Secure Board Configuration.

Update Procedure:

- HS-SE TIFS provides the capability to update SWREV values at runtime using the TISCI_MSG_WRITE_SWREV command.
- VPP_MCU is required for eFuse write operations.

Note

TI recommends NOT changing the SWREV of TIFS/SYSFW unless it is necessary. A wrong, non-existing SW revision for TIFS can brick the device.

4.2 KEYREV (Key Revision)

The KEYREV field in the OTP-eFuses controls which root of trust keys are used for secure boot operations:

- KEYREV = 1: Activates the Secondary Manufacturer Key Set (SMPK/SMEK)
- KEYREV = 2: Activates the Backup Manufacturer Key Set (BMPK/BMEK)

Update Procedure:

- KEYREV can be updated at runtime on HS-SE devices using the TISCI_MSG_WRITE_KEYREV command
- Updating KEYREV requires a dual-signed certificate with two components: Primary Certificate signed with SMPK private key and Secondary Certificate signed with BMPK private key
- TISCI_MSG_WRITE_KEYREV message can only be sent by host mentioned in the secure board config under write_host

For reference, see the FAQ on updating keyrev using TIFS API.

Copyright © 2026 Texas Instruments Incorporated

5 Common Debugging Steps

This section provides a systematic approach to diagnosing and resolving issues during key provisioning. Follow the steps for efficient troubleshooting.

5.1 Dry-Run Testing

- Remove/comment-out VPP-enable logic inside Keywriter application, build and run the keywriter tool
- If VPP is disabled, eFuse programming will fail.
- Refer to TIFS (Wakeup UART) traces for parsing validation and a failure message from keywriter TIFS as actual eFuse burning fails.

The following code shows the reference logs from WKUP and MCU UART on J721S2 EVM, command used for generating the cert blob is :

```
./gen_keywr_cert.sh -t ti_fek_public.pem
-s keys/smpk.pem --smek keys/smek.key -s-rp --smek-rp
-b keys/bmpk.pem --bmek keys/bmek.key -b-rp --bmek-rp
--mek-opt 0x1 --mpk-opt 0x21
--keycnt 2 --keycnt-wp
--keyrev 1
```


Copyright © 2026 Texas Instruments Incorporated

Table 5-1.

MCU UART (R5 app.)	WKUP UART (Keywriter TIFS)
OTP Keywriter Version: 02.00.00.00 (May 20 2026 - 16:02:05) OTP Keywriter ver: 10.1.4-v10.01.04a-1-gf7c27 (Fie Legacy Boot Mode Key programming sequence initiated Taking OTP certificate from 0x41c70004 Sciclient_otpProcessKeyCfg returns: -1 Debug response: 0x800 Key programming sequence completed	# Decrypting extensions.. # MPK Options: 0x21 MEK Options: 0x1 MPK Opt P1: 0x1 MPK Opt P2: 0x1 MEK Opt : 0x1 * SMPKH Part 1 BCH code: 61b45b59 * SMPKH Part 2 BCH code: 417d4cb4 * SMPK Hash (part-1,2): 1f6002b07cd9b0b7c47d9ca8d1aae57b8e8784a12f636b2b760d7d98a18f189701 60dfd0f23e2b0cb10ec7edc7c6edac3d9bdfefe0eddc3fff7fe9ad875195527d01 * SMEK BCH code: 21224fcc * SMEK Hash: 92785809a3dfefea57f6bbed642d730ba5d05e601222a72e815bf01ceb3a50f96ab85d282425f684436fabd4c7da624b791da411615035314103cc64e611f532 * BMPKH Part 1 BCH code: 81a4977b * BMPKH Part 2 BCH code: 613bf931 * BMPK Hash (part-1,2): b41865771ef3e13933568b1d1d50a72f5037a2a103c4535c713e10a61df1609c01 f6033117fbb56d5f190a17598e151141c6e413e2d853e56aae7997820639909b01 * BMEK BCH code: 21ec5cec * BMEK Hash: 21a5ae74b64b1c9fbd1f9201e60a319fe417edf06c2f9ba71fa3e1a967d171b684a4f85a83b8a4821abffe8e322b3697e299ffa94dec921bd4ac99f8ca59dd6c EXT OTP extension programming disabled MSV extension programming disabled * KEY CNT: 03030000 * KEY REV: 01010000 SWREV extension programming disabled FW CFG REV extension programming disabled * KEYWR VERSION: 0x20000 # # Programming Keys.. # * MSV: [u32] bch + msv: 0x8BAC0FFE MSV extension programming disabled [u32] bch + msv: 0x8BAC0FFE * SWREV: [u32] SWREV-SBL: 0x1 [u32] SWREV-SYSFW : 0x1 SWREV extension programming disabled [u32] SWREV-SBL: 0x1 [u32] SWREV-SYSFW : 0x1 * FW CFG REV: [u32] SWREV-FW-CFG-REV: 0x0 SWREV SEC BCFG extension programming disabled [u32] SWREV-FW-CFG-REV: 0x0 * EXT OTP: EXT OTP extension programming disabled * BMPKH, BMEK: Error in programming BMPKH part 1 debug_response: 0x800

Copyright © 2026 Texas Instruments Incorporated

5.2 Package Requirements

Verify the latest keywriter addon package is being used and is available from secure resources.

5.3 eFuse Programming Setup

VPP (Programming Voltage) must be stable throughout the key provisioning process.

5.4 Verifying Programming Results

- Successful eFuse programming returns debug response 0x0. Obtain the R5 keywriter application logs on MCU UART. The following reference logs show successful key provisioning.

```
OTP Keywriter Version: 02.00.00.00 (Jul 19 2023 - 11:15:35)
OTP Keywriter ver: 9.0.7-v09.00.07 (Kool Koala)
OTP_VppEn
WKUP_GPIO0_54 output high
Key programming sequence initiated
Taking OTP certificate from 0x41c7f004
Debug response: 0x0
Key programming sequence completed
```

- Non-zero responses indicate errors. See the TISCI Documentation for specific error code details.

5.5 Boot Failure Troubleshooting

If the device fails to boot after eFuse programming:

- Verify the key hash used to sign the boot image matches the hash programmed into OTP-eFuses.
- Extract the public key hash from the device and compare against Sec Cust MPK Hash (the active root of trust key hash).

  To extract the public key hash from the device, see the FAQ.

5.6 Trace Collection Without Wakeup UART

- Use the trace buffer to capture TIFS traces if wakeup UART is unavailable on the device.

Copyright © 2026 Texas Instruments Incorporated

6 Summary

This application note has provided a comprehensive overview of keywriter deployment strategies for secure device provisioning in production environments. Multiple deployment methods are available to accommodate different production scenarios, from automotive and industrial applications using OSPI boot to high-volume consumer electronics using eMMC boot.

Engineers must select the appropriate deployment method based on their specific production requirements, volume, and infrastructure. Proper understanding of certificate generation, verification processes, and common debugging steps verifies successful implementation and troubleshooting of keywriter operations in real-world environments.

7 References

1. Texas Instruments, *TISCI Key Writer User Guide*, user's guide.
2. Texas Instruments, *Jacinto7 High Security Device Development*, application note.
3. Texas Instruments, *PDK Keywriter User Guide*, user's guide.

Copyright © 2026 Texas Instruments Incorporated

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, regulatory 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 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, TI’s General Quality Guidelines, 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025