AN12800

i.MX RT10xx Fuse Provisioning for Security

Rev. 0 — 23 March 2020

Application Note

Contents

1 Introduction.. 1

2 Security lifecycle for i.MX RT10xx devices.. 1

3 Keys...2

3.1 Super Root Key Hash (SRK_HASH)........ 2

3.2 Bus Encryption Engine Key Selects (BEE_KEY0_SEL and BEE_KEY1_SEL).... 2

3.3 One-time Programmable Master Key (OTPMK).....2

3.4 Software General Purpose Key 2 (SW-GP2)..............................2

3.5 General Purpose 4 (GP4)............................ 3

4 JTAG/Debugging.................................. 3

5 Boot configuration.................................3

5.1 BOOT_CFG................. 3

5.2 BT_FUSE_SEL............ 3

5.3 DIR_BT_DIS................ 3

5.4 FORCE_COLD_BOOT.....................................4

6 Locks.. 4

7 Secure Fusing checklist.........................4

8 Reference... 8

9 Revision history.................................... 8

1 Introduction

NXP i.MX RT10xx series provides several security features most of which are controlled using fuses. For secure applications, there are some fuses that are not related to security features and might need to be configured. This document discusses fuse provisioning for secure applications and provides fuse configuration recommendations.

This application note assumes you are already familiar with the security features available on the i.MX RT10xx devices. For more information on security features, refer to the security reference manual for your i.MX RT device.

2 Security lifecycle for i.MX RT10xx devices

The security lifecycle for i.MX RT10xx devices has three states:

- Open (SEC_CONFIG[1] fuse = 0)
  - Intended for use in non-secure products or during the development phases of a secure product.
  - Signed images are optional. If a signed image is provided, authentication is performed and errors (if any) are logged, but authentication errors do not prevent boot.
  - SNVS transitions to Non-secure state during boot. SNVS master key is not available to the DCP module.
- Closed (SEC_CONFIG[1] fuse = 1; FIELD_RETURN fuse = 0):
  - Intended for use in secure products.
  - Signed images are mandatory. Non-authenticated code does not boot.
  - SNVS transitions to Trusted state during boot. SNVS master key is available to the DCP module as long as a security violation does not occur (for example, attaching a debugger).
- Field Return (SEC_CONFIG[1] fuse = 1; FIELD_RETURN fuse = 1):
  - Intended for the secure products that have been returned by the end customer. Device should not be returned to the service (cannot return to the Closed state).
  - Signed code with specific commands in the HAB CSF are required to allow transition from Closed to Field return.
    - Using HAB CSF unlock command including the device’s unique ID (CSF cannot be reused on another device), the field return sticky bit, which normally sets to block programming of the Field Return fuse, can be left clear.
    - After successful execution of the HAB CSF with the unlock field return command, the Field Return fuse can be blown to allow for additional debugging or sending the device back to NXP for analysis.

- — Signed images are optional. If a signed image is provided, authentication is performed and errors (if any) are logged, but authentication errors do not prevent boot.
- — JTAG can optionally be re-enabled if it has not been fully disabled using the SJC_DISABLE fuse (see Section 4 JTAG/Debugging for more information).

3 Keys

There are several keys and key select fuses available on the RT10xx devices. The following sections provide more detail on each of the keys. The SRK_HASH is the only value that is required to be programmed for secure applications. The other key values are optional depending on the use case.

3.1 Super Root Key Hash (SRK_HASH)

The SRK_HASH fuses must be blown with a value for secure applications running authenticated code. The value to program into the fuses corresponds to the public halves of the code signing key pairs that are used for the device. The SRK_HASH is used to authenticate the public key table that is appended to the signed image.

NOTE

Because the SRK_HASH is generated from public key information, read locking of the SRK is not needed.

3.2 Bus Encryption Engine Key Selects (BEE_KEY0_SEL and BEE_KEY1_SEL)

The Bus Encryption Engine (BEE) can be used for on-the-fly decryption all or portions of FlexSPI memory. The BEE module supports two regions where each can use a different key. If you plan to use the BEE module in your application, the BEE_KEY0_SEL and BEE_KEY1_SEL fuses must be properly configured to select the key that you want to use for each of the regions.

Even if you are not using the encrypted eXecute-In-Place (XIP) boot feature (configuring the BEE module in software after boot), the key selects still need to be configured as there is no field within the BEE module registers that allows selecting the key (the key selection only comes from the fuse block). The 0b00 option, which is a default option, for the key select is a “from register” option that allows programming a key value to use in the BEE registers. However, the “from register” fuse option must be selected to use this feature.

Other than the default “from register” option, the BEE keys that are available can vary across the RT10xx family. See the device-specific documentation for your device for more information.

NOTE

RT1010 devices use an OTFAD module instead of BEE. The OTFAD module has different key configuration options that are not described in this application note.

3.3 One-time Programmable Master Key (OTPMK)

The OTPMK is a device unique, that is, different on every processor, key that is used as a seed for key derivation. The OTPMK fuse value is programmed by NXP during chip manufacture. The key is also locked so that it cannot be read directly from fuses. The OTPMK value is sent to the SNVS module over a private bus. Then the OTPMK key value is derived and can be sent to the DCP or BEE module for use. The OTPMK-derived key can be used on the device but cannot be read even by NXP.

3.4 Software General Purpose Key 2 (SW-GP2)

The SW-GP2 fuse value can be a user-defined key, not provisioned by NXP, used by either the DCP or BEE module. If the SW-GP2 fuse is used as a key, write and read locking of the fuse using SW_GP2_LOCK and SW_GP2_RLOCK is recommended.

3.5 General Purpose 4 (GP4)

On some RT10xx devices, the GP4 fuse value can be used as a user-defined key, not provisioned by NXP, for the BEE module. If the GP4 fuse is used as a BEE key, write and read locking of the fuse using GP4_LOCK is recommended.

4 JTAG/Debugging

There are several JTAG and debugging modes available on the device and multiple fuses that control the operation. A secure application should never leave the debug fuses in the default state. Unlike some of the MCU products where enabling “secure” mode automatically disables debugging, on the RT10xx products, putting the device in Closed mode has no direct effect on debugging functionality. A closed device can be debugged although debugging triggers a security violation in the SNVS module.

To disable JTAG and debugging of the device completely and permanently, use the following fuse settings:

- JTAG_SMODE = 11, sets JTAG security mode to no debug
- SJC_DISABLE = 1, disables the JTAG module
- KTE = 1, disables trace
- JTAG_HEO = 1, disables HAB override of the JTAG security mode

For more information on using JTAG in secure mode, where debugging can be enabled using a challenge/response mechanism, see AN12419, “Secure JTAG for i.MX RT10xx.”

NOTE

When debugging is disabled, the JTAG_TRST signal can potentially interfere with software resets even if the debug communication is left at the default SWD setting. The JTAG_TRST pin should be pulled/driven low during software resets to avoid problems with the reset.

5 Boot configuration

The boot configuration fuses do not directly control security features, but their usage can have an impact on overall system security. The following sections describe how secure systems should provision the boot configuration fuses.

5.1 BOOT_CFG

The boot configuration for the device can be controlled using GPIO overrides or fuses. In order to save pins and to prevent an attacker from changing the boot configuration, secure applications should use the boot from fuses boot mode (BOOT_MODE[1:0] = 00). The BOOT_CFG fuses should be configured appropriately for your specific boot memory. NXP also recommends blowing the BOOT_CFG_LOCK fuse to prevent modification of the BOOT_CFG fuses after they have been set up.

NOTE

The BOOT_CFG fuse area contains fuses other than BOOT_CFG, so BOOT_CFG_LOCK (and other fuse locks) should be set as a final step in provisioning fuses after all other desired fuse configurations are complete.

5.2 BT_FUSE_SEL

When the BOOT_MODE[1:0] = 00 option is used to boot from fuses, the boot flow is controlled by the BT_FUSE_SEL value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, flash) is not programmed yet, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, the normal boot flow is followed, where the ROM attempts to boot from the selected boot device. Therefore, the BT_FUSE_SEL must be blown for normal boot operation with the BOOT_CFG values.

5.3 DIR_BT_DIS

The DIR_BT_DIS fuse should be blown to prevent use of reserved NXP functions. As of the writing of this application note, the fuse is already blown for RT106x devices when they leave the NXP factory. In future, this fuse might be blown during NXP manufacturing for other RT10xx products, but for now you must plan to burn this fuse during secure device provisioning.

5.4 FORCE_COLD_BOOT

RT10xx devices support a fast wakeup from Suspend low power mode option using SRC_GPR1[PERSISTENT_ENTRY0]. When this mechanism is used, most of the ROM is bypassed during the wake-up including code authentication (HAB). For the highest security, NXP recommends blowing the FORCE_COLD_BOOT fuse to prevent use of the PERSISTENT_ENTRY0 field.

6 Locks

In general, it is recommended to set as many of the documented lock fuses as possible in a final secure configuration to prevent malicious or unintended misuse. In particular, the BOOT_CFG_LOCK should be set to prevent modification of the boot setup. As mentioned,, if any of the optional fused keys are being used, read and write locking of the keys is also recommended.

7 Secure Fusing checklist

The table below provides a quick reference for fuse settings for secure applications. Not all of the fuses are required, but they should at least be reviewed and considered in determining the final fuse configuration.

Table 1. Secure Fusing Checklist

Security Lifecycle Fuses				
Fuse	Location	Recommended value for secure applications	Lock	Comment
SEC_CONFIG[1]	0x460[1]	1	BOOT_CFG_LOCK	Required for secure products. Should be one of the last fuses provisioned by OEM, followed by BOOT_CFG_LOCK.
FIELD_RETURN	0x400[31]	0	Protected by sticky bit in OCOTP controller: FIELD_RETURN_LOCK.	Should be 0 for normal operation of a secure product. A special HAB CSF command can be used to leave the field return sticky bit cleared. The Field Return fuse could then be blown.
Keys and key control fuses				
Fuse	Location	Recommended value for secure applications	Lock	Comment
SRK_HASH	0x580[31:0], 0x590[31:0], 0x5A0[31:0], 0x5B0[31:0], 0x5C0[31:0],	Derived from signing keys	SRK_LOCK. Write-protect and overwrite protect lock is only available on RT1010. For other RT10xx devices, there is no SRK_LOCK.	Required for secure products (needed for code signature authentication). The value to program into the fuses corresponds to the public halves of the code

Table continues on the next page...

Table 1. Secure Fusing Checklist (continued)

	0x5D0[31:0], 0x5E0[31:0], 0x5F0[31:0]			**signing key pairs that are used for the device.**
SRK_REVOKE[3:0]	0x6F0[3:0]	0	Protected by sticky bit in OCOTP controller: SRK_REVOKE_LOCK	Allows OEMs to manage root keys for HAB code signing by revoking selected keys. Each bit corresponds to an index in the SRK table.
BEE_KEY0_SEL	0x460[13:12]	Varies	BOOT_CFG_LOCK	Configure as needed to set the key for BEE region 0.
BEE_KEY1_SEL	0x460[15:14]	Varies	BOOT_CFG_LOCK	Configure as needed to set the key for BEE region 1.
OTPMK	N/A	Varies	N/A	NXP provisioned. Unique secret key programmed and locked by NXP. NXP or OEM cannot read the OTPMK.
SW-GP2	0x690[31:0], 0x6A0[31:0], 0x6B0[31:0], 0x6C0[31:0]	Varies	SW_GP2_LOCK and SW_GP2_RLOCK	Optional user provisioned key. Can be used as a key by the DCP and BEE modules.
GP4	0x8C0[31:0], 0x8D0[31:0], 0x8E0[31:0], 0x8F0[31:0]	Varies	GP4_LOCK and GP4_RLOCK	Available as a BEE key option on RT106x only.
JTAG and debugging fuses				
Fuse	Location	Recommended value for secure applications	Lock	Comment
JTAG_SMODE	0x460[23:22]	11	BOOT_CFG_LOCK	Secure applications should use the 01 mode to enable secure JTAG only or 11 to disable JTAG.

Table continues on the next page...

Table 1. Secure Fusing Checklist (continued)

				Note: JTAG can be reenabled using a special HAB CSF if SJC_DISABLE or JTAG_HEO is not blown.
SJC_DISABLE	0x460[20]	1	BOOT_CFG_LOCK	SJC_DISABLE fuse can optionally be blown to completely disable the secure JTAG controller.
KTE	0x460[26]	1	BOOT_CFG_LOCK	The KTE fuse can be blown to disable trace. If debug is disabled using JTAG_SMODE = 01 or 11, KTE must also be blown to disable debugging.
JTAG_HEO	0x460[27]	1	BOOT_CFG_LOCK	The JTAG_HEO fuse can optionally be blown to prevent reenabling of debugging using a HAB CSF.
Boot configuration fuses				
Fuse	Location	Recommended value for secure applications	Lock	Comment
BOOT_CFG1[7:0]	0x450[7:0]	Varies	BOOT_CFG_LOCK	Set as needed based on boot configuration.
BOOT_CFG2[2:0]	0x450[10:8]	Varies	BOOT_CFG_LOCK	Set as needed based on boot configuration.
BT_FUSE_SEL	0x460[4]	1	BOOT_CFG_LOCK	Blow BT_FUSE_SEL to use boot from fuses BOOT_MODE.
DIR_BT_DIS	0x460[3]	1	BOOT_CFG_LOCK	Set to disable NXP ROM test modes.
FORCE_COLD_BOOT	0x460[5]	1	BOOT_CFG_LOCK	Optionally set to disable fast suspend wake-up using persistent entry field as entry point, which bypasses HAB re-authentication on

Table continues on the next page...

Table 1. Secure Fusing Checklist (continued)

				suspend wake-up events.
Lock fuses				
Fuse	Location	Recommended value for secure applications	Lock	Comment
BOOT_CFG_LOCK	0x400[3:2]	11	N/A	Set to prevent modification of boot configuration fuses. Ensure that all BOOT_CFG fuses including SEC_CONFIG[1] are set as needed before blowing BOOT_CFG_LOCK.
SW_GP2_LOCK	0x400[21]	1	N/A	Optionally set to prevent modification of SW_GP2 key. Setting this lock is highly recommended if SW_GP2 key is used.
SW_GP2_RLOCK	0x400[23]	1	N/A	Optionally set to prevent software reading of SW_GP2 key. It does not block use of SW_GP2 by the DCP and BEE modules.
GP4_LOCK	0x400[25:24]	11	N/A	RT106x only. Optionally set to prevent modification of GP4 key. Setting this lock is highly recommended if GP4 key is used.
GP4_RLOCK	0x400[7]	1	N/A	RT106x only. Optionally set to prevent software reading of GP4 key. Does not block use of GP4 by the BEE module.
Other locks	Varies	Varies	N/A	Set documented lock bits for any fuses that you know will not require modification in the field.

8 Reference

- i.MX RT10xx Security Reference Manuals (moderated downloads)
- AN12419, “Secure JTAG for i.MX RT10xx”
- AN12681, “How to use HAB secure boot in i.MX RT10xx” (moderated download)
- AN12079, “How to use i.MXRT Security Boot” (moderated download)

9 Revision history

Table 2. : Revision history

Revision number	Date	Substantive changes
0	3/2020	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 23 March 2020

Document identifier: AN12800