AN14641

Fast and Secure Boot using Falcon Mode on i.MX 8M and i.MX 9

Rev. 1.0 — 24 April 2025

Application note

Document information

Information	Content
Keywords	AN14641, secure fast boot, Falcon mode, Falcon boot, secure boot, i.MX 8MN, i.MX 8MM, i.MX 8MP, i.MX 93, i.MX 95, Linux
Abstract	This document shows how to reduce the Linux secure-boot time on the i.MX 8M and i.MX 9 families, using the U-Boot Falcon mode.

1 Introduction

This document shows how to reduce the Linux secure-boot time on the i.MX 8M and i.MX 9 families, using the U-Boot Falcon mode. It is a slight variation of the method described in *Fast Boot on i.MX 8 and i.MX 9 Using Falcon Mode and Kernel Optimizations* (AN14093), enabling secure boot. The main difference is how and when the kernel device tree is fixed-up. In the original method, the device tree is manually fixed in U-Boot and saved for subsequent fast boot-ups. In the current method, the U-Boot SPL fixes the device tree at each boot. This process allows the use of a signed device tree and skips the manual step of fixing it.

2 Software and hardware environment

Software requirements:

- An Ubuntu 22.04 PC is assumed.
- This application note applies to the Yocto project BSP scarthgap release and Linux BSP release 6.6.36_2.1.0.

Hardware setup and equipment:

- Development kits:
  - NXP i.MX 8MM LPDDR4 EVK
  - NXP i.MX 8MN DDR4 EVK
  - NXP i.MX 8MP LPDDR4 EVK
  - NXP i.MX 93 11x11 LPDDR4 EVK
  - NXP i.MX 95 19x19 LPDDR5 EVK
- Cables:
  - Micro-USB for the debug port (i.MX 8M)
  - Type-C for the debug port (i.MX 9)
  - Type-C for the serial download port

3 A dive into the boot flow

This section describes how the system transitions from power-on reset to kernel execution. It also highlights the difference between the default boot, Falcon mode boot, and Secure Falcon mode boot.

3.1 Default boot

The boot ROM is the first program executed after a power-on reset. It handles the basic initializations for the bootloader (U-Boot) to start. Since U-Boot is too big to fit into the on-chip memory (OCRAM), it was divided into two parts: Secondary Program Loader (SPL) and U-Boot proper.

The SPL is a smaller preloader that runs from OCRAM. It initializes some peripherals, and the most importantly, the DRAM (on i.MX 95, the OEI initializes the DRAM). After initializing the DRAM, the SPL loads the ATF (ARM Trusted Firmware) and the U-Boot into the DRAM, then jumps to the ATF. Once ATF completes its tasks, it jumps to the U-Boot proper.

The U-Boot proper is the second stage bootloader. It provides a minimal set of tools to interact with the hardware through a command-line interface. The main tasks handled by the U-Boot include loading and preparing the kernel device tree (called 'fix-up') and the loading and starting the kernel image itself. The fix-up involves changing and adding some node parameters in the device tree, which includes the DRAM address and size, kernel boot arguments, and so on.

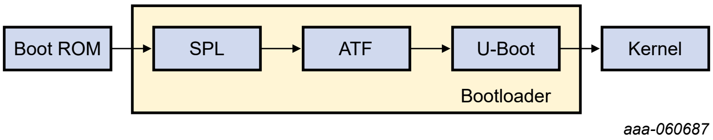

Figure 1. Default boot sequence

Note: *In i.MX 95, other steps are taken between boot ROM and SPL (OEI execution and System Manager boot). For clarity, those are omitted in this description. They can be considered part of the 'boot ROM' stage.*

3.2 Falcon mode boot

The Falcon mode is a U-Boot feature that allows loading and starting the Linux kernel directly from the SPL, bypassing the U-Boot proper. This mode enhances boot performance by reducing the steps involved in the default boot process.

The SPL handles both loading of the device tree and kernel, and starting of the kernel, eliminating the need for U-Boot proper. The device tree must be fixed-up in advance by either the SPL or through manual preparation. In *Fast Boot on i.MX 8 and i.MX 9 Using Falcon Mode and kernel Optimizations* AN14093, the device tree is fixed-up manually in U-Boot, and then saved on the boot device for the SPL to use. In this application note, the SPL fixes the device tree at runtime. This approach offers two-fold advantages: (1) the manual fix-up is not required, and (2) the device tree can be signed at compile time for secure boot and used as such.

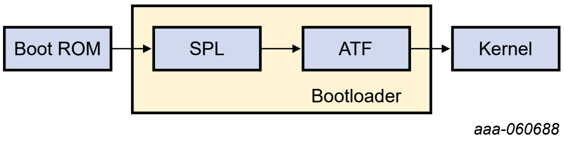

Figure 2. Falcon Mode boot sequence

3.3 Secure Falcon mode boot

The focus of this application note is the ability to implement secure boot. In the secure boot, each binary (SPL, ATF, device tree, and kernel) is signed with a set of cryptographic keys. During boot, each stage first verifies the signature of the subsequent stage, before starting it. This ensures that only authorized software is run on the platform, establishing a chain of trust. For more details related to the secure boot, check the Security Reference Manual of your platform, and the practical implementation details in the U-Boot repository.

In Falcon mode, the boot ROM authenticates the signature of the SPL before the boot ROM starts the SPL. The SPL loads the ATF, the device tree, and the Linux kernel. It authenticates the signatures for each of them, fixes the device tree, and then starts the ATF and the kernel.

4 How to run

To enable the Falcon mode with or without secure boot, follow the steps described in the README file of the source code. Make sure that the correct branch is selected according to the intended BSP release.

You can always fall back to U-Boot by keeping any key pressed during power on, on the serial console.

5 Implementation details

The implementation consists of several patches, described below.

- **The U-Boot patch**

In the meta-imx-fastboot/recipes-bsp/u-boot/files directory, there is a patch and a configuration file for each platform. For more details about the configured parameters, see the U-Boot documentation. Each 0001-<board>-add-falcon-mode-support.patch file:

– Implements the spl_start_uboot() function, located in uboot-imx/board/freescale/<board>/spl.c, where <board> is: imx8mm_evk, imx8mn_evk, imx8mp_evk, imx93_evk, or imx95_evk. This function checks if SPL should start the kernel or U-Boot. If any key is pressed during boot, the function returns 1, meaning that U-Boot must be started. Otherwise, SPL must start the kernel.

– The patch for i.MX 95 implements, in addition, the spl_fit_read() function in the arch/arm/mach-imx/imx9/scmi/soc.c file. Since the USDHC controller is a nonsecure controller, it cannot access the DDR secure region. This function is required only for i.MX 95 and it handles the container image loading from the storage device (SD or eMMC) to DDR.

The 0001-imx8m-reset-ethernet-phy-in-spl.patch file resets the Ethernet PHY for the i.MX 8M family. To bring it up in the operational state in which Ethernet MAC can interact with the PHY, this must be reset before starting the kernel. The PHY is reset in the board_init_r() function located in the uboot-imx/common/spl/spl.c file.

The 0001-fix-the-kernel-DTB-directly-in-SPL.patch file loads the device tree from the kernel FIT/container image and implements the fix-ups in SPL.

- **The ATF patch**

  In the meta-imx-fastboot/recipes-bsp/imx-atf/files directory, there is a patch for each platform. The patch adds support for jumping directly to the kernel. By default, the ATF is designed to jump to the U-Boot. To jump directly to the kernel on NXP platforms, the FDT address must be passed as an argument in the bl31_early_platform_setup2() function, located in imx-atf/plat/imx/imx8m/<board>/<board>_bl31_setup.c for i.MX 8M family and imx-atf/plat/imx/<board>/<board>_bl31_setup.c for i.MX 9 family.

- **The mkimage patch**

  The patches for the mkimage tool are located in the meta-imx-fastboot/recipes-bsp/imx-mkimage/files directory. Each 0001-<board>-add-falcon-mode-support.patch file:

  – Creates two new targets in the soc.mak file to generate:

    – The image containing the ATF, the kernel, and the kernel device tree; for the i.MX 8M family, an IVT header is added to the FIT image, to be signed.

    – The bootloader containing only the SPL.

  – In addition, the patch for the i.MX 8M creates the mkimage_fit_atf_kernel.sh script used for generating the FIT image source containing the ATF, the kernel, and the device tree. It adds the os property to the uboot-1 node of the U-Boot FIT image source (u-boot.its). This property is required when loading U-Boot (the case when spl_start_uboot() returns 1) while Falcon mode is enabled. Otherwise, the U-Boot fails to boot.

- **The kernel recipe append**

  The kernel boot arguments are added at compile time into the device tree, through the bbappend recipe file meta-imx-fastboot/recipes-kernel/linux/linux-imx_6.6.bbappend. To use custom kernel parameters, define the FALCON_KERNEL_BOOTARGS:<board> variable into the conf/layer.conf file. Check the README for an example of how to change the kernel parameters.

6 Benchmarks

This section presents, for reference, the timing results on our test boards. The measurements are based on the 6.6.36_2.1.0 BSP, with the image booting from eMMC. The measured interval is from reset to the first process in userspace.

Table 1. Booting time

Board	Default Boot		Fast Boot[1]	
	Nonsecure (ms)	Secure (ms)	Nonsecure (ms)	Secure (ms)
i.MX 8MN DDR4	7261	7662	1794	2218
i.MX 8MP	11013	11590	2270	2684
i.MX 8MM	8979	9929	4211[2]	5088[2]
i.MX 93	10126	12630	2326	4985
i.MX 95	16618	19276	3815[3]	6408[3]

[1] kernel log messages are suppressed using quiet.

[2] i.MX 8M Mini EVK does not come with an integrated Wi-Fi module connected to the PCIe port (unlike i.MX 8M Plus). Therefore, the PCIe PHY initialization consumes time, waiting for an active link. Also, the MMC UHS is not supported in SPL, increasing the loading time of the kernel image.

[3] DDR quick boot enabled.

Note: *eMMC fastboot mode is disabled in the current measurements. Enabling it could gain more speed.*

7 Revision history

Table 2 summarizes the revisions to this document.

Table 2. Revision history

Document ID	Release date	Description
AN14641 v.1.0	24 April 2025	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.