AN15075

Extending AHAB Authentication to Linux RemoteProc Firmware

Rev. 1.0 — 7 July 2026 Application note

Document information

Information	Content
Keywords	AN15075, Cortex-M33, i.MX 93, Secure Boot, Remote Proc
Abstract	This application note explains how to authenticate Cortex-M33 firmware loaded by the Linux remoteproc framework on i.MX 93.

1 Introduction

This application note explains how to authenticate Cortex-M33 firmware loaded by the Linux `remoteproc` framework on i.MX 93. The implementation packages the M33 ELF as an AHAB OEM container. ELE-AP authenticates the container and verifies the embedded image. The verified ELF payload then follows the normal remoteproc ELF flow.

The goal is to prevent Linux from loading arbitrary auxiliary-core firmware after boot. In a secure product, the bootloader and kernel are already authenticated by the primary AHAB boot chain. This design extends that trust requirement to M33 firmware loaded later through `/sys/class/remoteproc/remoteproc0/firmware`.

2 Authentication flow

The normal `remoteproc` path receives a firmware file. The path validates the file as a valid ELF and parses the resource table. The path then starts the remote processor.

The added AHAB path changes the firmware sanity step for i.MX `remoteproc`:

1. Linux receives the firmware filename through the `remoteproc sysfs` interface.
2. The i.MX `remoteproc` driver treats the file as an AHAB OEM container.
3. The container header is validated locally.
4. The complete container is copied into coherent memory.
5. ELE-AP authenticates the OEM container.
6. The driver copies the embedded ELF image into a fixed reserved memory region.
7. ELE-AP verifies the image hash and decrypts the payload when encryption is used.
8. The verified ELF payload replaces the firmware buffer.
9. After verification, the system proceeds with the standard `remoteproc` ELF validation and boot sequence.

The implementation depends on a reserved memory range that is visible to Linux and usable by ELE-AP verification. The image destination address generated by `imx-mkimage` must match the reserved memory address in the device tree.

3 Environment

Table 1. Environment

Item	Requirement
Board	i.MX 93 11x11 EVK
BSP	lf-6.6.36
Signing tool	CST 4.0.1
M33 firmware input	RPMsg ping-pong ELF for i.MX 93 11x11 EVK

The signing environment must contain AHAB PKI material, including an SRK table and an OEM signing certificate that matches the target fuse configuration. The encrypted flow also requires a 128-bit DEK.

4 Reference software package contents

The reference software package contains two source patches, licensing metadata, and validation logs.

Table 2. Package content

Component	Change
`imx-mkimage`	Adds `flash_remote_proc` to `iMX93/soc.mak`
`linux-imx`	Adds `CONFIG_IMX_REMOTEPROC_AHAB` and i.MX `remoteproc` AHAB handling
Device tree	Adds `remoteproc_reserved@a4220000`
Validation logs	A55 and M33 console captures

5 Kernel and image-tool integration

This section describes how to implement the feature in associated software.

5.1 Add the imx-mkimage target

Apply the `image-tool` patch.

```
cd imx-mkimage
git apply <reference_software_package>/imx-mkimage/0001-add-flash_remote_proc-
target-for-Linux-remote-proc-v.patch
```

The patch adds these defaults and target behavior.

```
REMOTE_PROC_IMAGE ?= m33_image.elf
REMOTE_PROC_ADDR ?= 0xA4220000
flash_remote_proc: $(MKIMG) $(REMOTE_PROC_IMAGE)
     ./$(MKIMG) -soc IMX9 -c -m33 $(REMOTE_PROC_IMAGE) 0 $(REMOTE_PROC_ADDR)
 $(REMOTE_PROC_ADDR) -out flash.elf
```

`REMOTE_PROC_ADDR` is the address used in the AHAB image metadata. It must remain synchronized with the Linux reserved memory node.

5.2 Enable kernel support

Apply the kernel patch.

```
cd linux-imx
git apply <reference_software_package>/linux-imx/0001-add-AHAB-check-for-imx-
remote-proc-process.patch
```

The patch enables:

```
CONFIG_IMX_REMOTEPROC_AHAB=y
```

It adds a 1 MB reserved memory node:

```
remoteproc_reserved: remoteproc_reserved@a4220000 {
    compatible = "shared-dma-pool";
    reg = <0 0xa4220000 0 0x100000>;
    no-map;
};
```

The patch also appends the reserved region to the `&cm33 memory-region` list, allowing the `remoteproc` driver to find and map it during probe.

5.3 ELE-AP command hooks

The kernel patch adds EdgeLock Secure Enclave (Advanced Profile) ELE-AP helper calls for the `remoteproc` firmware flow.

Table 3. ELE-AP commands

ELE-AP request	Purpose
`ELE_OEM_CNTN_AUTH_REQ`	Authenticate the OEM container header and signature block.
`ELE_VERIFY_IMAGE_REQ`	Verify the image hash for the selected image index.
`ELE_OEM_RELEASE_CNTN_REQ`	Release the authenticated container after verification.

The driver first maps the reserved memory region during probe and copies the embedded ELF payload into that region. Then, it verifies the image through ELE-AP, copies the verified payload back, and then calls the standard `rproc_elf_sanity_check()`.

5.4 Build and install kernel outputs

Build the kernel and install the image and DTB to the boot partition used by the target.

```
make imx_v8_defconfig
make -j32
cp arch/arm64/boot/Image /boot/
cp arch/arm64/boot/dts/freescale/imx93-11x11-evk.dtb /boot/
```

6 Building the authenticated M33 container

This section describes how to test the feature.

6.1 Create the unsigned container

Copy the M33 ELF into the `imx-mkimage/iMX93` directory using the expected filename and build the `remoteproc` container.

```
cd imx-mkimage/iMX93
cp imx93-11x11-evk_m33_TCM_rpmsg_lite_pingpong_rtos_linux_remote.elf
 m33_image.elf
cd ..
make SOC=iMX93 flash_remote_proc
```

The output is:

```
iMX93/flash.elf
```

The build log must provide the container and signature-block offsets used for CST signing.

```
CST: CONTAINER 0 offset: 0x0
CST: CONTAINER 0: Signature Block:
offset is at 0x90 Offsets =       0x0     0x90
```

6.2 Sign and encrypt

To sign and encrypt, perform the following steps:

1. Copy the generated container into the CST working directory and create a CSF file.
2. Replace the PKI paths with the signing material for the target platform.

```
[Header]
Target = AHAB
Version = 1.0

[Install SRK]
File = "<pki_dir>/SRK_1_2_3_4_table.bin"
Source = "<pki_dir>/SRK1_sha384_secp384r1_v3_usr_crt.pem"
Source index = 0
Source set = OEM
Revocations = 0x0

[Authenticate Data]
File = "flash.elf" Offsets = 0x0 0x90  [Install Secret Key]
Key = "encryption.key"
Key Length = 128
```

3. Run CST.

```
cst-4.0.1/bin/cst -i csf_remote_proc.csf -o signed-flash.elf
```

6.3 Append the DEK blob

To append the DEK blob, perform the following steps:

1. Generate a DEK blob on the target through U-Boot and append it to the signed container.

```
u-boot=> fatload mmc 1:1 0x80280000 encryption.key
u-boot=> dek_blob 0x80280000 0x80281000 128
u-boot=> fatwrite mmc 1:1 0x80281000 dek_blob.key 0x48
```

2. On the host or target shell, append the blob at the expected offset.

```
dd if=dek_blob.key of=signed-flash.elf bs=1 seek=704 conv=notrunc
```

7 Runtime loading from Linux

The validation flow first starts a default authenticated M33 image from U-Boot before Linux boots.

```
u-boot=> editenv mmcargs
edit: setenv bootargs ${jh_clk} ${mcore_clk} console=${console} root=${mmcroot}
 clk_ignore_unused
u-boot=> run prepare_mcore && fatload mmc 1:1 0x80000000 signed-
flash_bootaux_p384.bin && cp.b 0x80000000 0x201e0000 0x10000 && bootaux_cntr
 0x201e0000 0 && boot
```

After Linux boots, stop the current remote processor firmware and load the AHAB-protected replacement image.

```
echo 7 > /proc/sys/kernel/printk
echo stop > /sys/class/remoteproc/remoteproc0/state
echo -n /root/signed-flash.elf > /sys/class/remoteproc/remoteproc0/firmware
echo start > /sys/class/remoteproc/remoteproc0/state
```

The `remoteproc` firmware file can be supplied through the `sysfs` fallback path. The key requirement is that the file consumed by the driver is the signed AHAB container, not a raw ELF.

8 Validation strategy

To verify if the feature is running properly, check the following.

8.1 Confirm secure life cycle and reserved memory

U-Boot validation shows the target life cycle as OEM closed.

```
Lifecycle: 0x00000020, OEM closed
```

The Linux boot log must show the reserved verification memory.

```
OF: reserved mem: initialized node remoteproc_reserved@a4220000, compatible id
 shared-dma-pool
OF: reserved mem: 0x00000000a4220000..0x00000000a431ffff (1024 KiB) nomap non-
reusable remoteproc_reserved@a4220000
imx-rproc remoteproc-cm33: reserved mem: 0xa4220000 size 0x100000
```

8.2 Confirm successful authentication and boot

Expected A55-side output for a valid image:

```
imx-rproc remoteproc-cm33: OEM container /root/signed-flash.elf authenticate
 success
imx-rproc remoteproc-cm33: img->offset: 0x2000, img->size: 0xf000, img->dst:
 0xa4220000
imx-rproc remoteproc-cm33: OEM verify image success
remoteproc remoteproc0: Booting fw image /root/signed-flash.elf, size 69632
remoteproc remoteproc0: remote processor imx-rproc is now up
virtio_rpmsg_bus virtio1: creating channel rpmsg-openamp-demo-channel addr 0x1e
```

Expected M33-side output:

```
RPMSG Ping-Pong FreeRTOS RTOS API Demo...
RPMSG Share Base Addr is 0xa4010000
Link is up!
Nameservice announce sent.
```

8.3 Negative test

Use a modified or incorrectly signed container only on a system where an expected reset is acceptable.

```
echo stop > /sys/class/remoteproc/remoteproc0/state
echo -n /root/signed-flash_wrong.elf > /sys/class/remoteproc/remoteproc0/
firmware
echo start > /sys/class/remoteproc/remoteproc0/state
```

The validation evidence shows a reset with CSU as the reset reason after the tampered image is loaded on an OEM closed device.

```
Reset Status: CSU
```

9 Deployment guidance

The bootloader and kernel must be authenticated before relying on Linux `remoteproc` firmware authentication. Otherwise, an attacker could bypass the added driver logic.

Keep the container destination address, reserved memory address, and reserved memory size under configuration control. These values form a cross-component contract between `imx-mkimage`, the device tree, ELE-AP verification, and the `remoteproc` driver.

Validate both the successful firmware replacement path and the tampered-image path for the intended life cycle. OEM open and OEM closed life cycle behavior can differ, especially for failure handling.

10 Acronyms and definitions

Table 4. Acronyms and definitions

Acronym	Definition
AHAB	Advanced High Assurance Boot
BSP	Board Support Package
CST	Code Signing Tool
DEK	Data Encryption Key
DTB	Device Tree Blob
ELE-AP	EdgeLock Secure Enclave (Advanced Profile)
ELF	Executable and Linkable Format
EVK	Evaluation Kit
OEM	Original Equipment Manufacturer
PKI	Public Key Infrastructure
RPMsg	Remote Processor Messaging
SRK	Super Root Key
TCM	Tightly Coupled Memory

11 Related documentation

Table 5 lists and explains the additional documents and resources that you can refer to for more information.

Table 5. Related documentation

Document	Link / how to access
NXP i.MX 93 Applications Processor Reference Manual	IMX93RM
NXP i.MX Linux Reference Manual	RM00293
NXP i.MX Linux Yocto Project User's Guide	UG10164
NXP Code Signing Tool User Guide	CST Version 4.0.1

Table 5. Related documentation*...continued*

Document	Link / how to access
Linux kernel remoteproc framework documentation	Remote Processor Framework — The Linux Kernel documentation

12 Note about the source code in the document

Example code shown in this document has the following copyright and GPL-2.0 or (at your option) any later version license:

Copyright 2026 NXP

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see <https://www.gnu.org/licenses/>.

13 Revision history

Table 6 summarizes the revisions to this document.

Table 6. Revision history

Document ID	Release date	Description
AN15075 v.1.0	7 July 2026	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.

Contents