Abstract

This document explains how Manufacturing Protection works and describes it using it on a supported device.
1 Introduction

The section gives general information on i.MX RT1170 Manufacturing Protection.

Note: This application note was written using legacy tools and flow. The MCUXpresso Secure Provisioning (SEC) Tool and SPSDK are the latest tools. The information in this application note describing the secure flow within the chip still applies, but we recommend using the latest tools (MCUXpresso SEC or SPSDK) instead of following the steps shown here. For any questions, please contact your local support.

1.1 Purpose

The Manufacturing Protection feature available on i.MX RT1170 provides a mechanism to authenticate a device to an OEM. It intends to serve as the basis for establishing an authenticated communication channel used for the provisioning of sensitive data. It can be used to verify that a device has been fused with a correct configuration and has been properly secured to an OEM's specification in a remote facility. This document explains how the feature works and describes it using it on a supported device.

1.2 Audience and scope

This document is targeted in the i.MX RT1170 for users who want to understand:

- the manufacturing protection functionality available on the SoC
- the software enablement for this feature
- how to start developing an application that uses Manufacturing Protection

The reader must be familiar with basic secure provisioning and secure boot concepts.

Note: Devices from date codes prior to 2213 do not support this feature. For more information, visit the following link.

1.3 Acronyms and abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Client Applications</td>
</tr>
<tr>
<td>CAAM</td>
<td>Cryptographic Acceleration and Assurance Module</td>
</tr>
<tr>
<td>CSF</td>
<td>Command Sequence File</td>
</tr>
<tr>
<td>ECC</td>
<td>Elliptic Curve Cryptography (ECC) - uses the mathematical properties of elliptic curves to produce public key cryptographic systems</td>
</tr>
<tr>
<td>HAB</td>
<td>High Assurance Boot - a software library executed in the internal ROM on the NXP processor at boot time, which, among other things, authenticates software in external memory by verifying digital signatures in accordance with a CSF</td>
</tr>
<tr>
<td>KDF</td>
<td>Key Derivation Function - used to derive one (or more) keys from a secret value</td>
</tr>
<tr>
<td>MP</td>
<td>Manufacturing Protection</td>
</tr>
<tr>
<td>OTP</td>
<td>One-Time Programmable</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer - an organization that makes devices from component parts bought from other organizations</td>
</tr>
<tr>
<td>PKHA</td>
<td>Public Key Hardware Accelerator</td>
</tr>
</tbody>
</table>
Table 1. Acronyms and abbreviations...continued

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRK</td>
<td>Super Root Key - an RSA key pair that forms the start of the boot-time authentication chain. The hash of the SRK public key is embedded in the processor using OTP hardware. The SRK private key is held by the certificate authority.</td>
</tr>
<tr>
<td>TA</td>
<td>Trusted Application</td>
</tr>
<tr>
<td>ECDSA</td>
<td>Elliptic-Curve, Digital-Signature Algorithm</td>
</tr>
</tbody>
</table>

2 Overview

Most products that enable security require sensitive material to be installed at some point in their lifecycle. Provisioning secrets can be challenging to manage when a product is built by an untrusted contract manufacturer. The Manufacturing Protection feature available on NXP i.MX RT1170 processors provides an OEM with a tool to solve this problem. This feature is used when authenticating the SoC to the OEM's server.

To avoid overproduction and to keep track of how much the product was provisioned, the OEM provisioning server must create a record of all the existing devices using the message received from the device when the provisioning process started. The message should contain the unique ID of the device or its serial number.

Table 2. OEM manufacturing concerns and Manufacturing Protection objectives

<table>
<thead>
<tr>
<th>OEM manufacturing concerns</th>
<th>Manufacturing Protection objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Access to product-specific</td>
<td>• Preventing access to product secrets</td>
</tr>
<tr>
<td>secret keys or encrypted</td>
<td>• Detecting the “overproduction” of devices</td>
</tr>
<tr>
<td>boot keys</td>
<td>• Providing a method to authenticate NXP processors</td>
</tr>
<tr>
<td>• Overproduction</td>
<td>• No need for special security environments at a contract manufacturing</td>
</tr>
<tr>
<td>• Product cloning</td>
<td>facility</td>
</tr>
</tbody>
</table>

2.1 Manufacturing Protection keys

The foundation for the Manufacturing Protection feature is an ECC private/public key pair generated by the SoC. The private key is unreadable and unexportable. It is coupled to the OEM SRK fuse data used for a secure boot and it is only available for use after a device is enabled for a secure boot.

The Cryptographic Acceleration and Assurance Module (CAAM) supplies the key generation capabilities and the High Assurance Boot (HAB) manages whether the Manufacturing Protection feature is available during the system boot. The CAAM offers the following four functions to support the feature:

1. The first is a function to generate a non-readable and non-exportable private key (MPPrivK).
2. The second is a function to sign data with the private key (MPSign).
3. The third function generates a corresponding public key to be used for authenticating the signatures generated by devices (MPPubK).
4. The fourth function generates a shared secret value that can be used as a symmetric cryptographic key, or as a secret input to a key derivation function (MP-ECDH).

All the Manufacturing Protection functions must use the same ECC domain curve. This is specified via the Curve Select field in the function’s Protocol data block. An error is generated if different curves are specified in different Manufacturing Protection functions. Therefore, the ROM code generates the MPPrivK using a predefined ECC curve. The ECC curve set in the CAAM descriptors used for MPSign, MPPubK, and MP-ECDH generation must be the same. In the process of verifying the MP signature, the same ECC domain curve must be used. In the i.MX RT1170, the ROM predefined ECC curve is P256.

The details about building the descriptors and the supported ECC domain curves are in the i.MX RT1170’s security reference manual.
2.2 Manufacturing Protection private key generation

There are several fixed secret internal key modifiers supplied as inputs to the private key generation function. These secrets are uniform for a given part number. For example, all i.MX processors with the same part number contain the same SoC-specific secrets to generate the private key. In addition to these static device secrets, the OEM SRK fuse configuration is supplied as a key modifier. It ensures that the private key is also bound to the OEM SRK used for the device secure boot and only the OEM authenticated software should be able to use the key for signing purposes.

The key derivation modifiers are as follows:

- NXP 'part number uniform 'secret
- OEM super root key fuse configuration (image-signing keys)

After a device is configured for the secure boot, the HAB generates the MP private key during each boot cycle. The processor only generates keys once per power-on-reset.

After running the MPPrivK-generation function, the CAAM block stores the ECC curve that was selected when the MPPrivK generation protocol was executed in the CAAM SCFGR register. Unless a specially crafted signature is supplied during the boot, the HAB clears the private key to prevent potential misuse. It also makes it possible to make the private key available only for the provisioning software.

2.3 Manufacturing Protection sign

MPSign is the elliptic-curve, digital-signature algorithm (ECDSA) signing function used in the manufacturing protection authentication process. MPSign supports only ECDSA in prime fields. This function takes message data as input and outputs a signature over a message composed of the content of the MPMR, followed by the input-data message.

2.4 Manufacturing Protection public key generation

The MPPubk_generation function uses the private key value stored in the MPPrivK register by the MPPrivk_generation function to generate a matching elliptic-curve DSA public key. The curve selected via the Csel field in the PDB must match the curve used by the MPPrivk_generation function, or else an error is generated. The public key created by the MPPubk_generation function is written out to the specified results destination. The MPPubK Generation function is intended to be run once, at the OEM facility, but no harm is done if it is run at other times.
2.5 Manufacturing Protection Diffie-Hellman function

MP-ECDH is the elliptic-curve Diffie-Hellman function used in the manufacturing protection authentication process. MP-ECDH is used for the key agreement between a chip and another entity. The MP-ECDH function takes as an input the public key of the other entity, and, using the private key generated by the MPPrivk_generation function, generates a shared secret value. This same shared secret value can be produced by the other entity using the other entity's private key and the chip's public key, which is generated by the MPPubk_generation function. This shared secret value could be used as a symmetric cryptographic key, or it could be used as a secret input to a key derivation function.

One possible use for the shared secret value is in downloading encrypted software or data to the chip. Because the shared secret can be generated only by the chip and the other entity, only these two parties are able to encrypt or decrypt data protected via that shared secret.

3 Using Manufacturing Protection

This section describes the basic steps required to enable and deploy a system using the Manufacturing Protection feature.

3.1 Enabling secure boot

The first step to using Manufacturing Protection is to enable the secure boot feature. When a device is enabled for the secure boot, the Manufacturing Protection private key is generated automatically by the SoC during the boot. The secure boot process is out of the scope of this document. For details on how to enable the secure boot, see section How to use i.MX RT Security Boot in AN12079.
3.2 Private key persistence

The next step is to ensure that the private key is available to the software after the device boots. The Manufacturing Protection private key is cleared during the boot unless the bd file contains the 'Unlock' command, informing the HAB to leave the key. The 'Unlock' command is added to the bd description file, as shown below.

```plaintext
options {
    flags = 0x08;
    startAddress = 0x30000000;
    ivtOffset = 0x1000;
    initialLoadSize = 0x2000;
    entryPointAddress = 0x30002000;
}
sources {
    elfFile = extern(0);
}

constants {
    SEC_CSF_HEADER               = 20;
    SEC_CSF_INSTALL_SRK          = 21;
    SEC_CSF_INSTALL_CSFK         = 22;
    SEC_CSF_INSTALL_NOCAK        = 23;
    SEC_CSF_AUTHENTICATE_CSFK    = 24;
    SEC_CSF_INSTALL_KEY          = 25;
    SEC_CSF_AUTHENTICATE_DATA    = 26;
    SEC_CSF_INSTALL_SECRET_KEY   = 27;
    SEC_CSF_DECRYPT_DATA        = 28;
    SEC_NOP                      = 29;
    SEC_SET_MID                  = 30;
    SEC_SET_ENGINE               = 31;
    SEC_INIT                     = 32;
    SEC_UNLOCK                   = 33;
}

section (SEC_CSF_HEADER;
    Header_Version="4.5",
    Header_HashAlgorithm="sha256",
    Header_Engine="ANY",
    Header_EngineConfiguration=0,
    Header_CertificateFormat="x509",
    Header_SignatureFormat="CMS"
)
{

}

section (SEC_CSF_INSTALL_SRK;
    InstallSRK_Table="crts/SRK_1_2_3_4_table.bin", // "valid file path"
    InstallSRK_SourceIndex=0
)
{
}

section (SEC_CSF_INSTALL_CSFK;
    InstallCSFK_File="crts/CSF1_1_sha256_4096_65537_v3_usr_crt.pem", // "valid file path"
    InstallCSFK_CertificateFormat="x509" // "x509"
)
{
}

section (SEC_CSF_AUTHENTICATE_CSFK)
{
}
```
Note: Unless the private key is preserved during the boot, none of the Manufacturing Protection features are functional.

3.3 Exporting the public key

During product development, the public key must be generated and saved for use by an authentication or provisioning server. The public key is used to remotely verify all the Manufacturing Protection signed data. The public key can be generated using the example MPPubk_generation that is supported in caam.c.

To guarantee the integrity of the public key, which is critical for proper use of the Manufacturing Protection, the OEM should generate the public key within an OEM-trusted facility.

In the following example, the key is generated using the example public key generation function called RunMPPubKExample():

Figure 4. Output logs of example MPPubk_generation function

**Figure 3. Exporting a public key to a provisioning server**

GOAL: Boot securely and generate the Production MP key pair. Export the Public Key so the Provisioning server can verify signatures created using the Private key.
3.4 Enabling a provisioning image

A newly assembled product at a contract manufacturing facility must execute a software image capable of establishing a communication channel with the OEM authentication/provisioning server. This software image must be signed with the appropriate 'Unlock' command to enable Manufacturing Protection. The provisioning image then generates messages and signs them using the MPSign function. The authenticity of the device is established when the signed message is received on the server and verified with the Manufacturing Protection public key. A secure connection can then be established for sensitive data provisioning on the new product.

![Diagram showing chip distribution and manufacturing protection](image)

Figure 5. Chip distribution with Manufacturing Protection

The messages originating from the new target are signed using the MPSign function provided by the CAAM. The example MPSign function RunMPSignExample() demonstrates the signing of a data package:

```plaintext
MPSign example:
MPSign using P256 curve:
Message: 0x44566777
Each of signed messages: 5CE044677179311BC17D31F65741968D45784B3B0C83279F52226C685
MP signature data is:
59: B0180645C402A27B8B6411B999E35BCC2C4C0D2FCA2205D084AB0F3F0526
MP signature size: 256 bit
generate signature success!
```

Figure 6. Output logs of example MPSign function

These signed messages can be useful not only for authenticating and establishing a secure connection, but also for verifying certain aspects of the target. For example, system designers can modify the messages to contain things like device UIDs, a copy of the fused configuration of the SoC, software versions.

3.5 MP public key – key derivation function

The scope of the Manufacturing Protection feature is to provide a mechanism to authenticate a device to an OEM. It serves as the basis for establishing an authenticated communication channel used for sensitive data provisioning. However, if the product is assembled at the contract manufacturing site and the OEM has secret data to be installed on a product that does not support a communication interface, the MP public key can be used to seed a KDF to derive a symmetric encryption key.
The Manufacturing Protection public key is exported and stored on the OEM secure server. On the server, the MP public key is used to derive the symmetric encryption/decryption key that is used to encrypt the product secrets. Finally, the encrypted secrets are included in the software image.

On the device, the MP public key seeds a KDF and recreates the symmetric encryption/decryption key. The encrypted secrets are decrypted using the derived key and stored securely on the device using CAAM blobs.

In this use case, the MP public key is not public and must be treated as a secret shared with the OEM secure server. The provisioning software must not output the MP public key. During the production process, the RunMPPubKExample() function must be removed to prevent the MP public key leakage.

3.5.1 Private key software management

It is recommended that the software image clears the Manufacturing Protection private key when it is no longer needed for signing purposes. The key can be cleared by writing a bit in the CAAM Security Configuration Register (SCFGR). See the respective security reference manual for details.

**Note:** The MP private key cannot be cleared after the secure keys (JDKEK, TDKEK, and TDSK) are generated.

3.6 MP-ECDH – generate shared secret

CAAM also implements a fourth Manufacturing Protection function called MP-ECDH. It can be used for Manufacturing Protection authentication with, or in lieu of, the public key signature authentication process. The MP-ECDH function generates a shared secret between a CAAM-equipped device and an external entity. This shared secret can be used for decrypting confidential data (for example, keys, software) downloaded to the device from the external entity. The external entity can be assured that only a genuine NXP part, configured correctly, and running in the proper security state can generate this shared secret. The authenticity of the external entity’s public key must be ensured by the software that initiates the MP-ECDH function. One way to authenticate the public key would be to include it in the signed image whose signature is verified in the boot process. One advantage of the MP-ECDH-based authentication process is that there is no need for the device to contact an external server. The confidential data to be decrypted with the Manufacturing Protection shared secret could be prepared in advance and distributed with the signed boot image.

In this use case, the shared secret is not public. The MP-ECDH function takes as inputs the public key of the other entity and the private key generated by the MPPrivk_generation function:
4 Software enablement

The Manufacturing Protection functionality can be enabled using an attachment that contains fsl_caam.c, fsl_caam.h, and caam.c. See the latest AN release available on www.nxp.com.

**Note:** These functions only work correctly when the device is configured for the secure boot and when the boot image signature contains the ‘Unlock’ command.

5 References

The following documents may offer further reference.

- i.MX Manufacturing Protection (document AN13222)
- How to use i.MX RT Security Boot (document AN12079)
- Security Reference Manual for the i.MX RT1170 Processor (document i.MX RT1170 SRM)
- Secure the Edge: Manufacturing Protection: Provision Sensitive Material in an Unsecure Environment (training Manufacturing Protection)

6 Revision history

Table 3 summarizes the revisions to this document.

<table>
<thead>
<tr>
<th>Revision number</th>
<th>Release date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21 September 2023</td>
<td>The document is updated to correspond to the latest guidelines, Section 1 is updated.</td>
</tr>
<tr>
<td>0</td>
<td>27 June 2022</td>
<td>Initial public release</td>
</tr>
</tbody>
</table>
7 Legal information

7.1 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.

7.2 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. 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’s responsibility is 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 http://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 data sheet 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.

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.

7.3 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.

i.MX — is a trademark of NXP B.V.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong> Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Purpose</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Audience and scope</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Acronyms and abbreviations</td>
<td>2</td>
</tr>
<tr>
<td><strong>2</strong> Overview</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Manufacturing Protection keys</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Manufacturing Protection private key generation</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Manufacturing Protection sign</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Manufacturing Protection public key generation</td>
<td>4</td>
</tr>
<tr>
<td>2.5 Manufacturing Protection Diffie-Hellman function</td>
<td>5</td>
</tr>
<tr>
<td><strong>3</strong> Using Manufacturing Protection</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Enabling secure boot</td>
<td>5</td>
</tr>
<tr>
<td>3.2 Private key persistence</td>
<td>6</td>
</tr>
<tr>
<td>3.3 Exporting the public key</td>
<td>7</td>
</tr>
<tr>
<td>3.4 Enabling a provisioning image</td>
<td>8</td>
</tr>
<tr>
<td>3.5 MP public key – key derivation function</td>
<td>8</td>
</tr>
<tr>
<td>3.5.1 Private key software management</td>
<td>9</td>
</tr>
<tr>
<td>3.6 MP-ECDH – generate shared secret</td>
<td>9</td>
</tr>
<tr>
<td><strong>4</strong> Software enablement</td>
<td>10</td>
</tr>
<tr>
<td><strong>5</strong> References</td>
<td>10</td>
</tr>
<tr>
<td><strong>6</strong> Revision history</td>
<td>10</td>
</tr>
<tr>
<td><strong>7</strong> Legal information</td>
<td>11</td>
</tr>
</tbody>
</table>