EC2x&EG2x&EG9x Series

Secure Boot Application Note

LTE Standard Module Series

Version: 1.0

Date: 2023-06-30

Status: Released

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm.
Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2023. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2023-05-16	Eve HU/ Damon LI	Creation of the document
1.0	2023-06-30	Eve HU/ Damon LI	First official release

Contents

About the Document **3**
Contents **4**
Table Index **5**
Figure Index **6**

1 Introduction **7**
1.1. Applicable Module 7

2 Secure Boot Overview **8**
2.1. Definition 8
2.2. Flow Chart 8

3 Secure Boot Enabling Procedures **10**
3.1. Certificate Chain Generation 10
3.1.1. Environment Preparation 10
3.1.2. Certificate and Keys Generation 10
3.1.3. Certificate and Key Replacement 11
3.2. QFPROM Configurations 11
3.2.1. Parameter Configuration 11
3.2.2. sec.dat File Generation 12
3.3. Image Signing 13
3.3.1. Modules with Flash Memory in 128 M 13
3.3.2. Modules with Flash Memory in 256 M or 512 M 14
3.4. Secure Firmware Downloading 15

4 FAQs **17**
4.1. Do I Need to Execute Certificate Chain Repeatedly 17
4.2. Do I Need to Generate sec.dat Repeatedly 17
4.3. Can Unsigned Firmware Run on Module 17
4.4. Can Original Firmware Be Downloaded After Enabling Secure Boot 17
4.5. How to Perform Verification After Enabling Secure Boot 17
4.6. Does Secure Boot Still Work After Erasing sec Partition 18

5 Appendix References **19**

Table Index

Table 1: Applicable Module 7
Table 2: Terms and Abbreviations 19

Figure Index

Figure 1: Secure Boot Flow Chart 8
Figure 2: Generate Certificates and Keys 10
Figure 3: Certificates and Keys 11
Figure 4: Configure QFPROM Parameter 12
Figure 5: sec.dat File 12
Figure 6: Prepare Images to Be Signed 13
Figure 7: Sign Images 13
Figure 8: Signed Images List 14
Figure 9: Prepare Images to Be Signed 14
Figure 10: Sign Images 14
Figure 11: Signed Images List 15
Figure 12: Firmware Package with sec.dat File 15

1 Introduction

This document explains how to enable Secure Boot on Quectel EC2x family, EG2x family and EG9x family modules.

1.1. Applicable Module

Table 1: Applicable Module

Module Family	Module
EC2x	EC20-CE
	EC21 Series
	EC25 Series
EG2x	EG21-G
	EG25-G
	EG21-GL
	EG25-GL
EG9x	EG91 Series
	EG95 Series

NOTE

This document is applicable to the above modules and their QuecOpen® solutions.

2 Secure Boot Overview

2.1. Definition

Secure Boot refers to the secure bootup sequence that establishes a trusted platform by incorporating signature verification. It ensures that only authorized software is executed during the boot process. Secure Boot system incorporates signature verification to each stage of the module booting process to prevent the execution of any unauthorized or maliciously modified software on the module.

2.2. Flow Chart

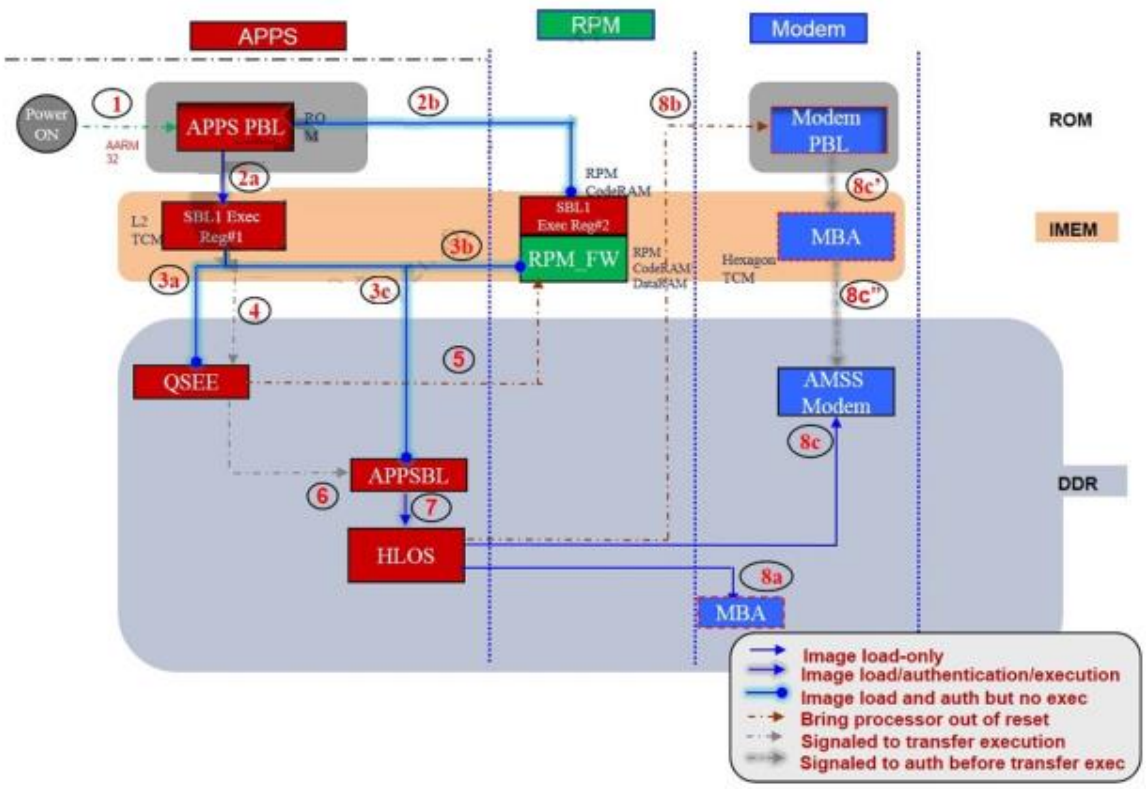


Figure 1: Secure Boot Flow Chart

NOTE

According to the image signature format, Secure Boot is divided into two parts:

- Classic Secure Boot mechanism that does not support signing HLOS (Kernel and FS) images.
- Aboot (Bootloader) authentication with the authentication of Kernel (*mdm9607-perf-boot.img*). The public key certificate used for authentication is stored in the *aboot* file.

3 Secure Boot Enabling Procedures

3.1. Certificate Chain Generation

3.1.1. Environment Preparation

1. Software environment: Ubuntu system, Python 2.7, OpenSSL 1.0.2 or later versions.
2. Contact Quectel Technical Support for the corresponding tool installation packages *gencerts.zip* for generating certificates and keys as well as *ql_9207_nork_mro_SecBootTools_Release_v1.2m.tar.gz* used for generating signed images.

3.1.2. Certificate and Keys Generation

To generate the certificates and keys:

Step 1: Open Ubuntu terminal window and unzip *gencerts.zip*
Step 2: Execute **./genCerts.sh** to generate certificates and keys.

```
gary@q-PowerEdge-R730:~/work/secboot$ unzip gencerts.zip
Archive:  gencerts.zip
   creating: certs/
  inflating: certs/genCerts.sh
  inflating: certs/v3.ext
  inflating: certs/opensslroot.cfg
  inflating: certs/v3_attest.ext
gary@q-PowerEdge-R730:~/work/secboot$ cd certs/
gary@q-PowerEdge-R730:~/work/secboot/certs$ ./genCerts.sh
 Quectel Wireless Solutions Co., Ltd. All rights reserved
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Ignoring -days; not generating a certificate
Signature ok
subject=C = CN, ST = ANHUI, L = HEFEI, O = QUEC, OU = OPEN, CN = oem_rootca
Getting Private key
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            0f:a1:44:e5:29:40:a8:49:a8:52:18:96:02:e3:53:12:d7:8d:56:53
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = CN, ST = ANHUI, L = HEFEI, O = QUEC, OU = OPEN, CN = oem_rootca
        Validity
            Not Before: May 19 01:29:38 2021 GMT
            Not After : May 14 01:29:38 2041 GMT
        Subject: C = CN, ST = ANHUI, L = HEFEI, O = QUEC, OU = OPEN, CN = oem_rootca
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
```

Figure 2: Generate Certificates and Keys

If the command is executed successfully, the generated certificates and keys will be stored under the */certs* of *gencerts.zip* directory.

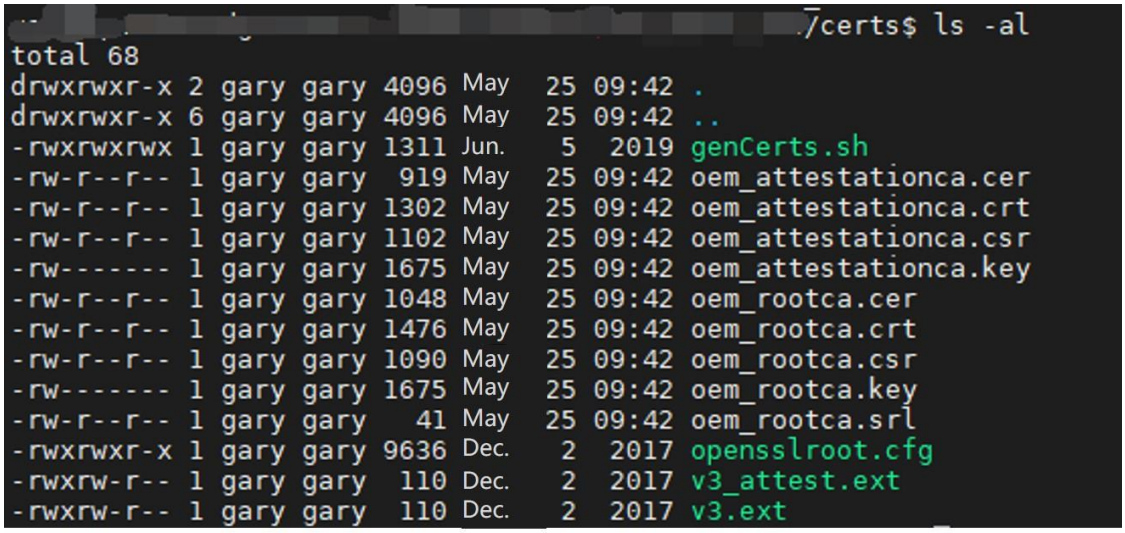


Figure 3: Certificates and Keys

3.1.3. Certificate and Key Replacement

To replace the certificates and keys:

Step 1: Unzip the *ql_9207_nork_mro_SecBootTools_Release_v1.2m.tar.gz* tool package.

Step 2: Copy the newly generated *oem_rootca.key*, *oem_rootca.cer*, *oem_attestationca.key* and *oem_attestationca.cer* files to the *ql_9207_nork_mro_SecBootTools_Release_v1.2m/cert* directory to replace the existing files in the directory with the newly copied files.

3.2. QFPROM Configurations

QFPROM, which stores configurations related to chip authentication in Non-Volatile ROM, can achieve the safe environment required by Secure Boot. This chapter provides an overview of the parameter configurations that must be set before signing images.

3.2.1. Parameter Configuration

To configure the parameters:

Step 1: Open the *9607_fuseblower_USER.xml* file in the *common/sectools/config/9607* directory of *ql_9207_nork_mro_SecBootTools_Release_v1.2m* package.

Step 2: Locate the *root_cert_hash* parameter within the file and replace the existing value with the new value. Note that the other option values typically remain at their default settings.



Figure 4: Configure QFPROM Parameter

NOTE

The value of *root_cert_hash* is obtained by signing *oem_rootca.cer* file with SHA256 algorithm. To obtain the value, execute **sha256sum oem_rootca.cer** in Ubuntu terminal window.

3.2.2. sec.dat File Generation

Execute **genSecdat.sh** in Ubuntu terminal window to generate *sec.dat*. The file will be stored in the *output/SecDat* of *ql_9207_nork_mro_SecBootTools_Release_v1.2m* directory.

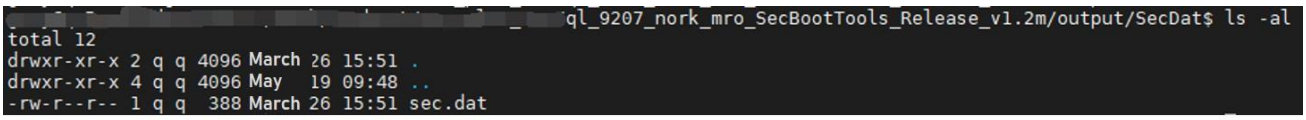


Figure 5: sec.dat File

3.3. Image Signing

3.3.1. Modules with Flash Memory in 128 M

1. Unzip the latest firmware package, replace all images to be signed in the of *ql_9207_nork_mro_SecBootTools_Release_v1.2m/firmwares* directory with the signed versions:

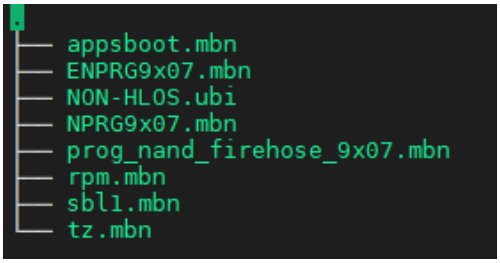


Figure 6: Prepare Images to Be Signed

2. Copy *ql-rootfs.tar.gz* file from the *secboot* folder in the firmware package to the *ql_9207_nork_mro_SecBootTools_Release_v1.2m/firmwares* directory.

3. Follow the instructions in the *README* file of *ql_9207_nork_mro_SecBootTools_Release_v1.2m* to create a Docker environment. Run the *signImage.sh* script of *ql_9207_nork_mro_SecBootTools_Release_v1.2m* in the newly created Docker environment to sign the images.

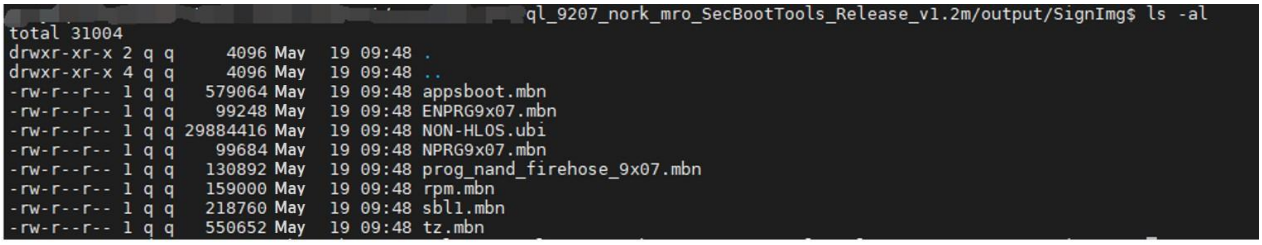


Figure 7: Sign Images

4. The signed images will be stored in *ql_9207_nork_mro_SecBootTools_Release_v1.2m/output/SignImg*.

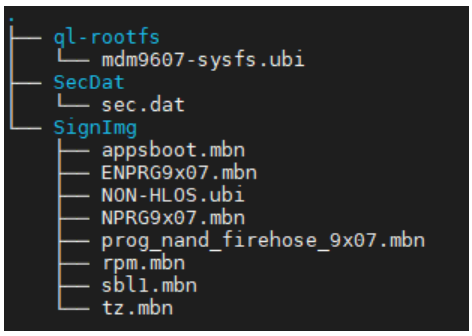


Figure 8: Signed Images List

NOTE

When you copy the singed images from the *SignImg* folder to the *update* directory in the firmware package, you also need to copy the *mdm9607-sysfs.ubi* in the *ql-rootfs* folder.

3.3.2. Modules with Flash Memory in 256 M or 512 M

1. Unzip the latest firmware package, and replace all images to be signed in the *ql_9207_nork_mro_SecBootTools_Release_v1.2m/firmwares* directory with the signed ones:

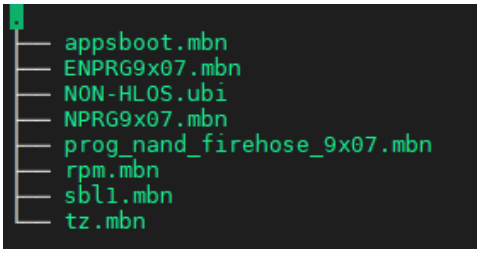


Figure 9: Prepare Images to Be Signed

2. Follow the instructions in the *README* file of *ql_9207_nork_mro_SecBootTools_Release_v1.2m* to create a Docker environment. Run the *signImage.sh* script of *ql_9207_nork_mro_SecBootTools_Release_v1.2m* in the newly created Docker environment to sign the images.

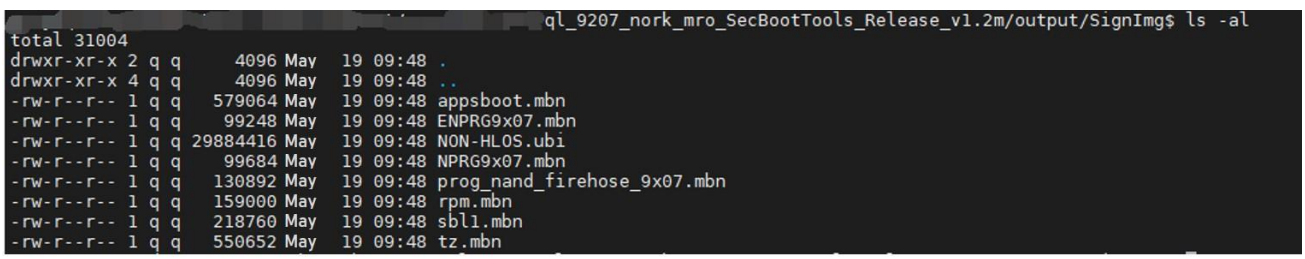


Figure 10: Sign Images

3. The signed images will be stored in *ql_9207_nork_mro_SecBootTools_Release_v1.2m/output/SignImg*.

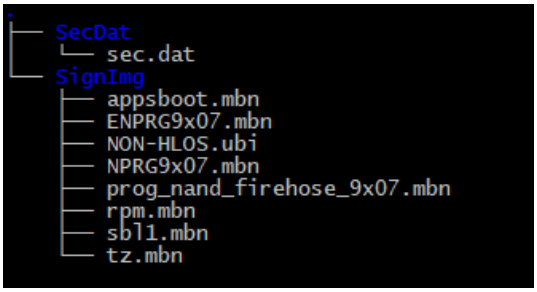


Figure 11: Signed Images List

3.4. Secure Firmware Downloading

1. The generated firmware package includes the signed images and *sec.dat,* as shown below. Among them, the *sec.dat*, *patition.mbn* and *partion_nand.xml* files are used to enable Secure Boot without signed images.

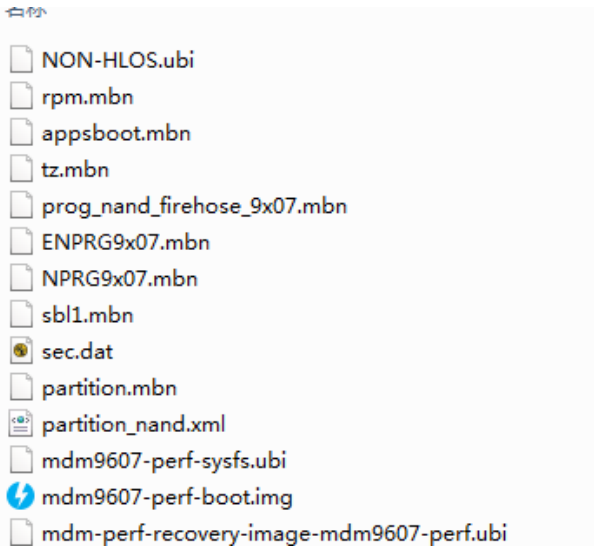


Figure 12: Firmware Package with sec.dat File

2. Download the firmware package without *sec.dat* to verify if the module can be booted normally.

3. Once the module is booted successfully, download *sec.dat* separately. Before downloading, you must modify *rawprogram_nand_p2K_b128K_update.xml* file (for modules with 128 M or 256 M flash memory) or *rawprogram_nand_p4K_b256K_update.xml* (for modules with 512 M flash memory) in the *update/firehouse/* directory of the firmware package to support downloading the *sec* partition. Contact Quectel Technical Support for the *rawprogram_nand_p2K_b128K_update.xml* or *rawprogram_nand_p4K_b256K_update.xml* file.

NOTE

Secure Boot must be enabled during the module production stage. Enabling Secure Boot after upgrading the module to a version that supports Secure Boot is strictly prohibited, as it may result in abnormal functioning and irrecoverable issues. No configuration, such as SIMLock configuration, should be performed before enabling Secure Boot.

4 FAQs

4.1. Do I Need to Execute Certificate Chain Repeatedly

If the keys and certificates remain unchanged, there is no need to execute certificate chain repeatedly. The certificate chain only needs to be generated once. For subsequent firmware upgrades, you only need to re-sign the images (see ***Chapter 3.3*** for details).

4.2. Do I Need to Generate sec.dat Repeatedly

If the module model and root certificate remain unchanged, there is no need to re-generate *sec.dat*.

4.3. Can Unsigned Firmware Run on Module

Before enabling Secure Boot, unsigned firmware can run on the module. However, once Secure Boot is enabled, unsigned firmware cannot run on the module.

4.4. Can Original Firmware Be Downloaded After Enabling Secure Boot

You can only download the original firmware with fastboot method.

4.5. How to Perform Verification After Enabling Secure Boot

Perform verification by downloading the original incorrectly signed firmware. Once Secure Boot is enabled, firmware that is not properly signed cannot be downloaded.

4.6. Does Secure Boot Still Work After Erasing sec Partition

Erasing sec partition does not affect the functionality of Secure Boot.

5 Appendix References

Table 2: Terms and Abbreviations

Abbreviation	Description
HLOS	High Level Operating System
IoT	Internet of Things
QFPROM	Qualcomm Fuse Programmable Read Only Memory
ROM	Read Only Memory