RG50xQ&RM5xxQ Series

Secure Boot Application Note

5G Module Series

Version: 1.1

Date: 2023-07-27

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.

About the Document

Revision History

Version	Date	Author	Description
-	2021-03-22	Ritchie WU	Creation of the document
1.0	2021-04-02	Ritchie WU	First official release
1.1	2023-07-27	Shaun DUAN/ Jayde TONG	1. Added the following applicable modules: RG502Q-EA, RG502Q-EU and RG502Q-GT. 2. Deleted the applicable module RM502Q-GL. 3. Updated the related information of certificate chain (Chapter 2.3). 4. Added AT+QSECBOOT="roothash" (Chapter 3.3.1.3). 5. Updated the content of Characteristic in AT+QSECBOOT="progsec" (Chapter 3.3.1.4). 6. Updated Matters Needing Attention (Chapter 4).

Contents

Table Index

1 Introduction

Quectel 5G RG50xQ series and RM5xxQ series modules support Secure Boot function. This document describes how to use AT commands to enable the Secure Boot function on RG50xQ series and RM50xQ series modules, including Secure Boot overview, detailed explanations of AT commands, and matters needing attention.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Family	Module Series	Model
RG50xQ	RG500Q	RG500Q-CN/-EA/-EU/-GT
	-	RG501Q-EU
	RG502Q	RG502Q-EA/-EU/-GT
RM5xxQ	RM500Q	RM500Q-AE/-CN/-GL
	-	RM502Q-AE
	-	RM505Q-AE
	-	RM510Q-GL

2 Secure Boot Overview

2.1. Secure Boot Definition

Secure Boot refers to a secure boot-up sequence that established on a trusted platform. Signature verification is additionally required in Secure Boot during the booting process to ensure that only the verified software can be executed.

At each stage of the module booting process, signature verification needs to be additionally performed in Secure Boot to prevent any software without valid signature or maliciously modified software from running on the module. A root trusted entity is needed during the booting process. Primary Boot Loader (PBL) is embedded in RG50xQ series and RM50xQ series modules as a firmware, which is unmodifiable, and therefore can serve as the root trusted entity.

2.2. Secure Boot Enabling

Secure Boot can only be enabled with the fuse on the hardware and cannot be disabled after being enabled.

The module booting process comprises multiple stages. One dedicated image in each stage performs a specific function. After enabling Secure Boot, the image in each stage needs to be verified by the previous image before execution. If the verification cannot be passed, the entire booting process fails, and the module fails to boot up.

As the root trusted entity, PBL (also known as RoT) is the firmware embedded in chips and therefore cannot be modified. Therefore, the PBL is considered as the most trusted entity in the booting process and used to perform authentication in the next booting stage. Normally, SBL is verified in the second booting stage and it is executed after it has been successfully authenticated by the PBL. Since the SBL has been trusted, the SBL can be used to authenticate the image in the next stage.

2.3. Certificate Chain

Secure Boot adopts the ECSDA algorithm, secp384r1 curve, and SHA-384 digest algorithm for signing the certificates and images.

The certificate chain of the module adopts three-level certificates: Self-signed root certificate, Attestation CA certificate and Attestation certificate.

2.4. Image Signing

The standard format of images is ELF. During Secure Boot, to verify each stage of booting process, images of each stage need to be signed. A binary file in the standard ELF format includes several segments indicating different types of information separately, wherein the *hash table segment* stores signature related information. *Hash table segment* also includes the hash values of each segment and information about certificate trust chain.

The images that must be signed in the module are as follows:

- *abl.elf*
- *aop.mbn*
- *devcfg.mbn*
- *hyp.mbn*
- *multi_image.mbn*
- *prog_firehose_sdx55.mbn*
- *sbl1.mbn*
- *tz.mbn*
- *uefi.elf*
- *xbl_cfg.elf*
- *apdp.mbn*
- *NON-HLOS.ubi*

2.5. QFPROM Configuration

RG50xQ series and RM5xxQ series modules include one-time programmable fuses. The initial states of all fuses are 0 (which indicates that the Secure Boot is not enabled). Once a writing operation is performed on the fuse (or the fuse is blown), the state of the fuse permanently becomes 1 (which indicates that the Secure Boot is enabled). The state cannot be changed after the fuse is blown. To start Secure Boot, the program tool QFPROM is required.

QFPROM is used to store, in NVROM (non-volatile read-only memory), configurations related to chip authentication and can implement the secure environment required by Secure Boot. Configure QFPROM to blow the fuse, and then to implement all required security functions, such as output of Debug port, JTAG, secure file system and prevention of software rollback .

3 Secure Boot Related AT Commands

3.1. AT Command Syntax

3.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

3.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 2: Types of AT Commands

Command Type	Syntax	Description
Test Command	**AT+<cmd>=?**	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of the corresponding command.
Write Command	**AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]**	Set user-definable parameter value.
Execution Command	**AT+<cmd>**	Return a specific information parameter or perform a specific action.

3.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about the use of the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendations or suggestions about how to design a program flow or what status to set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there is a correlation among these examples, or that they should be executed in a given sequence.

3.3. AT Commands Description

3.3.1. AT+QSECBOOT Enable or Query Secure Boot

AT+QSECBOOT Enable or Query Secure Boot	
Test Command **AT+QSECBOOT=?**	Response **+QSECBOOT: "status",(**list of supported **<enable>**s**)** **+QSECBOOT: "serialnum",<serial_number>** **+QSECBOOT: "roothash",<root_hash>** **+QSECBOOT: "progsec",(**list of supported **<enable>**s**)** **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristic	-

3.3.1.1. AT+QSECBOOT="status" Query Enabling Status of Secure Boot

This command queries whether Secure Boot is enabled in the module.

AT+QSECBOOT="status" Query Enabling Status of Secure Boot	
Write Command **AT+QSECBOOT="status"**	Response **+QSECBOOT: "status",<enable>** **OK** If there is any error:

	ERROR
Maximum Response Time	300 ms
Characteristic	-

Parameter

<enable>	Integer type. Enabling status of Secure Boot. 1 Secure Boot is enabled 0 Secure Boot is not enabled

3.3.1.2. AT+QSECBOOT="serialnum" Query the Unique Serial Number of the Module

This function queries the unique serial number of the module.

AT+QSECBOOT="serialnum"	**Query the Unique Serial Number of the Module**
Write Command **AT+QSECBOOT="serialnum"**	Response **+QSECBOOT: "serilalnum",<serial_number>** **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristic	-

Parameter

<serial_number>	String type. Serial number of the module in hexadecimal format without double quotes.

3.3.1.3. AT+QSECBOOT="roothash" Query the Hash Value of the Root Certificate

This command queries the hash value of the root certificate after Secure Boot is enabled in the module.

AT+QSECBOOT="roothash" Query the Hash Value of the Root Certificate	
Write Command **AT+QSECBOOT="roothash"**	Response **+QSECBOOT: "roothash",<root_hash>** **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristic	-

Parameter

<root_hash>	String type. Hash value of the root certificate in hexadecimal format without double quotes.

NOTE

The hash value of the root certificate is 0 when Secure Boot is not enabled.

3.3.1.4. AT+QSECBOOT="progsec" Enable Secure Boot

This function enables Secure Boot.

AT+QSECBOOT=" progsec " Enable Secure Boot	
Write Command **AT+QSECBOOT="progsec"[,<enable>]**	Response If the optional parameter is omitted, query the current setting: **+QSECBOOT: "progsec",<enable>** **OK** If the optional parameter is specified, enable Secure Boot . **OK**

	If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristic	This command takes effect after the module is rebooted. The configuration is saved automatically.

Parameter

<enable>	Integer type. Enable Secure Boot or not. 1 Enable Secure Boot 0 Secure Boot is not enabled (Only valid in the query result.)

4 Matters Needing Attention

1. Secure Boot can only be enabled with the hardware fuse and cannot be disabled after being enabled.

2. It is recommended to use **AT+QSECBOOT="progsec",<enable>** to enable Secure Boot, which is not enabled by default. This command downloads the image file in the *sec* partition and automatically activates the Secure Boot after rebooting the module.

3. After the Secure Boot is enabled, it is not supported to restore the firmware to a version that does not support the Secure Boot through Firehose.

4. After the Secure Boot is enabled, it is not supported to restore the firmware to a version that does not support the Secure Boot through FOTA, otherwise the module cannot be booted normally.

5. The PCIe Fuse mode also downloads the image file in the *sec* partition, which conflicts with the enabling of Secure Boot. If the Secure Boot is enabled first, the PCIe Fuse mode cannot be enabled. Therefore, you need to enable PCIe Fuse mode first and then enable the Secure Boot.

6. Secure Boot must be enabled at the factory stage. It is forbidden to enable the Secure Boot after the firmware version is upgraded to a version that supports the Secure Boot. Otherwise, some functions may be abnormal and cannot be recovered. Before the Secure Boot is enabled, it is strongly recommended not to perform any configuration on the module, such as SIMLOCK configuration.

5 Appendix Terms and References

Table 3: Terms and Abbreviations

Abbreviation	Description
CA	Certificate Authority
ECSDA	Elliptic Curve Digital Signature Algorithm
ELF	Executable and Linkable Format
FOTA	Firmware Over-The-Air
PBL	Primary Boot Loader
PCIe	Peripheral Component Interconnect Express
QFPROM	Qualcomm Fuse Programmable Read Only Memory
ROM	Read Only Memory
RoT	Root of Trust
SBL	Secondary Boot Loader
SHA	Secure Hash Algorithm