BG96 Secure Boot Application Note

LPWA Module Series

Version: 1.0

Date: 2022-02-18

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
-	2021-12-22	Justice HAN	Creation of the document
1.0	2022-02-18	Justice HAN	First Official Release

Contents

Table Index

1 Introduction

This document describes Secure Boot details and how to enable Secure Boot on Quectel BG96 module.

2 Secure Boot Overview

2.1. Definition

Secure Boot is defined as a boot sequence in which each firmware image to be loaded and executed is authorized using the previously authorized firmware.

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

2.2. Secure Boot Process

The Secure Boot process comprises multiple stages, and the image in each stage performs a specific function. After the Secure Boot is enabled, the image to be executed in each stage needs to be verified by the previously verified image. If the verification fails, the entire boot process stops, and the module cannot boot up. Quectel BG96 module follows the verification sequence of Primary Boot Loader (PBL) → Secondary Boot Loader (SBL) → ARM® TrustZone.

- As the root of trust (RoT), the PBL is the firmware embedded in chips and cannot be modified. Therefore, it is considered as the most trusted entity in the boot process, and authenticates the image to be executed in the next boot stage.
- The SBL is usually verified in the second boot stage. After it is successfully authenticated by the PBL, it can be executed and used to authenticate the image in the next stage.

NOTE

Secure Boot is disabled by default. For details on how to enable Secure Boot, see ***Chapter 3***.

2.3. Certificate Chain

Secure Boot supports 2048-bit RSA public keys with exponent 3 or F4 (= 65537) for signatures of the certificates and images. The format of the certificate signatures meets the *PKCS #1 v1.2* standard and the SHA256 or SHA384 algorithm.

The certificate chain of the module supports two-level certificate chain which includes attestation certificate and self-signed root certificate. The X.509 v1, v2 and v3 certificate formats are all supported.

2.4. Image Signing

During Secure Boot, the images to be executed in each boot stage must be signed first. Quectel firmware images use the standard MBN format, and each image includes several segments indicating different types of information separately, wherein the hash table segment stores signature related information. The hash table segment also includes the hash values of each segment and the information about certificate trust chain.

The images listed below must be signed in the Secure Boot process for Quectel BG96 modules.

- *apps.mbn*
- *mba.mbn*
- *qdsp6sw.mbn*
- *ENPRG9x06.mbn*
- *NPRG9x06.mbn*
- *rpm.mbn*
- *sbl1.mbn*
- *tz.mbn*

2.5. Hardware Foundation

Quectel BG96 module includes a one-time programmable fuse. The initial state of the fuse is 0 (Secure Boot disabled). Once a writing operation is performed on the fuse (or the fuse is blown), the state of the fuse permanently becomes 1 (Secure Boot enabled). The state cannot be changed after the fuse is blown, which means that the Secure Boot enabling is an irreversible operation.

2.6. Secure Boot Toolkit

Quectel provides a Secure Boot toolkit (Quectel SecBootTools) to generate related certificates and the *sec.dat* file, and to sign firmware images. The following document introduces the directory structure of Quectel SecBootTools in Windows system.

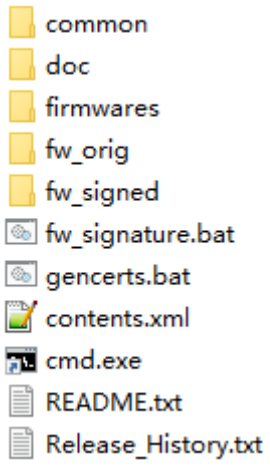

Figure 1: Quectel SecBootTools Directory Structure

Table 1: Quectel SecBootTools Directory Structure

SN	File/Folder	Description
1	*common*	Contains the toolchain for signature.
2	*doc*	Contains all reference documents on Secure Boot.
3	*firmwares*	Stores firmware packages to be signed and secure firehose configuration files.
4	*fw_orig*	An intermediary for creating signed firmware packages.
5	*fw_signed*	Stores signed firmware packages and intermediate files.
6	*fw_signature.bat*	Makes a signed firmware package.
7	*gencerts.bat*	Generates the root certificate (*qpsa_rootca.cer*) and the attestation certificate (*qpsa_attestca.cer*) as well as the hash values.
8	*contents.xml*	A configuration file used during signature. It cannot be modified.
9	*README.txt*	For more details about the above files or folders, refer to *README.txt*.
10	*Release_History.txt*	Stores the release history of the tool.

NOTE

Contact Quectel Technical Support (support@quectel.com) to acquire the Secure Boot toolkit.

2.7. sec.dat

The *sec.dat* file is vital for enabling Secure Boot, as it includes the configuration parameters for the following functions,.

1. Secure Boot enabling
2. JTAG access disabling
3. Anti-rollback enabling
4. Read/Write permissions disabling/enabling for fuses
5. Fuse blowing

The *sec.dat* file is generated during the image signing procedure. See ***Chapter 3.1.4*** for details.

3 Enable Secure Boot

3.1. Procedure

3.1.1. Preparation

Store the original firmware package to be signed under *fw_orig*.

Then, install Python and OpenSSL and check if paths of Python and OpenSSL defined in */common/scripts/env.bat* are the same as the actual paths.

3.1.2. Generate Certificates and Public Key Hash Value

Run *gencerts.bat* in the Secure Boot toolkit to generate a root certificate and an attestation certificate, as well as a public key hash value of the root certificate. The generated certificates are automatically stored in */fw_signed/output/certs*, and they are used to sign images; and the hash value is used to verify the signed images. If any image does not pass verification, the loading of the image will fail.

3.1.3. Fill in Environment Variables

Open *fw_signature.bat* and fill in the environment variables defined in *fw_signature.bat* with the firmware version to be signed.

3.1.4. Sign Images and Generate a Firmware Package

Run *fw_signature.bat* in the toolkit to sign the necessary image files and generate a new firmware package. For the list of necessary images, see ***Chapter 2.4***.

The whole process includes:

1) Double click *fw_signature.bat* and then *fw_signature.bat* runs automatically to enter Stage 1. In Stage 1, *sec.dat* which contains the hash of the root CA is automatically generated and stored in */fw_signed/output/sec_dat*.
2) Press any key to proceed to Stage 2, during which the necessary image files are signed one by one. Signed files are automatically stored in */fw_signed/output/9206tx*.
3) Press any key to proceed to Stage 3, during which a new firmware package is automatically created

in *fw_signed* and replaces the original image files with the signed image files of the same names.

4) Press any key to proceed to Stage 4, during which the firehose configuration file *rawprogram_nand_p2K_b128K_update.xml* in the new firmware package is replaced with a secure firehose configuration file *rawprogram_nand_p2K_b128K_sec.xml*, and *partition_nand.xml* is replaced with a secure-exclusive one.

3.1.5. Flash Firmware

Flash the firmware with the signed package created in ***Chapter 3.1.4***. For details about how to flash firmware, see ***document [1]***.

3.2. Verification

After firmware updating, send **AT+QSECBOOTSTAT?** to query whether Secure Boot is enabled on the module.

3.2.1. AT+QSECBOOTSTAT Query Secure Boot Status

This command queries the current status of Secure Boot.

AT+QSECBOOTSTAT Query Secure Boot Status	
Read Command **AT+QSECBOOTSTAT?**	Response **+QSECBOOTSTAT: <status>** **OK** If there is any error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	-

Parameter

<status>	Integer type. Secure Boot status. 0 Disabled 1 Enabled

Example

```
AT+QSECBOOTSTAT?          //Query whether Secure Boot is enabled on the module.
```

```
+QSECBOOTSTAT: 1          //Secure Boot is enabled.

OK
```

4 Appendix References

Table 2: Related Documents

Document Name
[1] Quectel_QFlash_User_Guide

Table 3: Terms and Abbreviations

Abbreviation	Description
ARM	Advanced RISC Machine
CA	Certificate Authority
JTAG	Joint Test Action Group
PBL	Primary Boot Loader
PKCS	Public-Key Cryptography Standards
RoT	Root of Trust
RSA	Algorithm invented by Rivest, Adleman and Shamir
SBL	Secondary Boot Loader
SHA	Secure Hash Algorithm