BG95&BG77&BG600L Series Secure Boot Application Note

LPWA Module Series

Version: 1.1

Date: 2021-12-30

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.

About the Document

Revision History

Version	Date	Author	Description
-	2020-11-03	Harding HU	Creation of the document
1.0	2020-12-02	Harding HU	First official release
1.1	2021-12-30	Justice HAN	1. Added two images (*cmnlib.mbn* and *keymasterapp32.mbn*) that must be signed in Secure Boot process (Chapter 2.4). 2. Updated the directory structure of Secure Boot toolkit (Chapter 2.6). 3. Updated Secure Boot enabling procedure (Chapter 3.1).

Contents

Table Index

1 Introduction

This document describes Secure Boot details and how to enable Secure Boot on Quectel BG95 series, BG77 and BG600L-M3 modules.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Series	Model	Description
BG95	BG95-M1	Cat M1
	BG95-M2	Cat M1/Cat NB2
	BG95-M3	Cat M1/Cat NB2/EGPRS
	BG95-M4	Cat M1/Cat NB2, 450 MHz Supported
	BG95-M5	Cat M1/Cat NB2/EGPRS, Power Class 3
	BG95-M6	Cat M1/Cat NB2, Power Class 3
	BG95-MF	Cat M1/Cat NB2, Wi-Fi Positioning
BG77	BG77	Cat M1/Cat NB2
BG600L	BG600L-M3	Cat M1/Cat NB2/EGPRS

NOTE

Hereinafter, BG95 series is collectively referred to as BG95 unless otherwise specified.

2 Secure Boot Overview

2.1. Definition

Secure Boot is defined as a boot sequence in which each loaded and executed firmware image 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 each image in every stage performs a specific function. After enabling the Secure Boot, 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 BG95, BG77 and BG600L-M3 modules follow the verification sequence of Primary Boot Loader (PBL) → Secondary Boot Loader (SBL) → ARM® TrustZone.

- As the root of trust, the PBL (also known as RoT) 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 or 4096-bit RSA private keys for signatures of the certificate and images. The format of the certificate signatures meets the *PKCS #1 v1.5* or *ITU-T X.509 v3* Standard and the SHA1 or SHA256 algorithm.

The certificate chain of Quectel BG95, BG77 and BG600L-M3 modules consists of two certificates in X.509 format based on SHA-384 algorithm: the self-signed root certificate and the attestation certificate.

The required certificates can be generated through the *gencerts.bat* mentioned in ***Chapter 2.6***. The generated certificates are used to sign the image and the *sec.elf* file (see ***Chapter 2.7*** for details).

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 ELF 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 BG95, BG77 and BG600L-M3 modules.

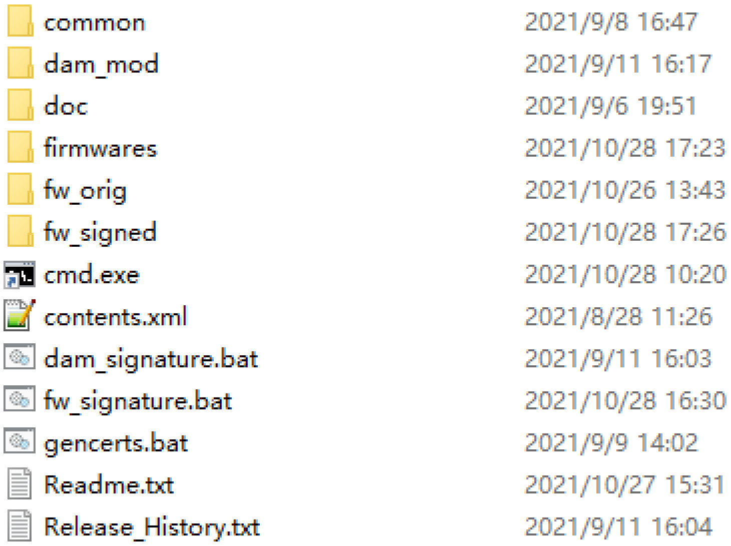
- *sbl1.mbn*
- *prog_firehose_nand_mdm9x05.elf*
- *tz.mbn*
- *devcfg.mbn*
- *rpm.mbn*
- *multi_image.mbn*
- *qdsp6sw.mbn*
- *qdsp6sw_2.mbn*
- *apps.mbn*
- *cmnlib.mbn*
- *keymasterapp32.mbn*

2.5. Hardware Foundation

The modules include one-time programmable fuses. The initial state of all fuses is 0 (Secure Boot disabled). Once a write operation is performed on the fuse (or the fuse is blown), the fuse state permanently becomes 1 (Secure Boot enabled). The state cannot be changed after the fuse is blown, which means that 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.elf* file, and to sign software images. The following introduces the directory structure of Quectel SecBootTools in Windows system.



common	2021/9/8 16:47
dam_mod	2021/9/11 16:17
doc	2021/9/6 19:51
firmwares	2021/10/28 17:23
fw_orig	2021/10/26 13:43
fw_signed	2021/10/28 17:26
cmd.exe	2021/10/28 10:20
contents.xml	2021/8/28 11:26
dam_signature.bat	2021/9/11 16:03
fw_signature.bat	2021/10/28 16:30
gencerts.bat	2021/9/9 14:02
Readme.txt	2021/10/27 15:31
Release_History.txt	2021/9/11 16:04

Figure 1: Quectel SecBootTools Directory Structure

Table 2: Quectel SecBootTools Directory Structure

SN	File/Folder	Description
1	*common*	Contains the toolchain for signature.
2	*dam_mod*	Stores QuecOpen application images to be signed and secure firehose configuration files for different projects.
3	*doc*	Contains all reference documents on Secure Boot.
4	*firmwares*	Stores firmware package to be signed and secure firehose configuration files for different projects.
5	*fw_orig*	An intermediary for creating signed firmware packages.
6	*fw_signed*	Stores the signed firmware package and intermediate files.

7	*contents.xml*	A configuration file used during signature. It cannot be modified.
8	*dam_signature.bat*	Signs QuecOpen application images.
9	*fw_signature.bat*	Makes a signed firmware package.
10	*gencerts.bat*	Generates the root certificate (*qpsa_rootca.cer*) and the attestation certificate (*qpsa_attestca.cer*) as well as the hash values.
11	*Release_History.txt*	Stores the release history of the tool.
12	*Readme.txt*	For more details about the above files or folders, refer to *Readme.txt*.

NOTE

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

2.7. sec.elf

The *sec.elf* file is vital for enabling Secure Boot, as it includes the configuration parameters for the function, as illustrated below.

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.elf* 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 Python and OpenSSL paths 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, image loading 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 and corresponding project name.

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.elf* that contains the hash of the root CA is automatically generated and stored in */fw_signed/output/sec_elf*.
2) Press any key to proceed to Stage 2, during which the necessary images are signed one by one. Signed images are automatically stored in */fw_signed/output/9205*.
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 images with the signed images of the same names.

4) Press any key to proceed to Stage 4, during which the firehose configuration file *rawprogram_nand_p2K_b128K.xml* in the new firmware package is replaced with the secure firehose *rawprogram_nand_p2K_b128K_sec.xml*.

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]***.

NOTE

After the Secure Boot is enabled, if you download an unsigned image, or use certificates different from the ones used for enabling Secure Boot to sign the image, the module will fail in downloading the firmware during firmware flashing process.

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 3: Related Document

Document Name
[1] Quectel_QFlash_User_Guide

Table 4: Terms and Abbreviations

Abbreviation	Description
CA	Certificate Authority
ELF	Executable and Linkable Format
JATG	Joint Test Action Group (an industry standard for verifying designs and testing printed circuit boards)
PBL	Primary Boot Loader
PKCS	Public-Key Cryptography Standards
RoT	Root of Trust
RSA	Rivest–Shamir–Adleman
SBL	Secondary Boot Loader
SSL	Secure Sockets Layer