MT793X_Secure_eFuse_Writer_User_Guide

Version: 0.4

Release date: 2022-07-26

Version History

Version	Date	Author	Description
0.1	2021-11-05	SS Wu	Initial draft
0.2	2021-11-15	Ryan Wu	Re-version
0.3	2022-01-10	Ryan Wu	Add efuse info
0.4	2022-07-26	Ryan Wu	Add efuse info

Table of Contents

List of Tables

List of Figures

1 MT793X Use Command to Read/Write Secure eFuse

1.1 Introduction

Mediatek provides an “eFuse library” which is independent from FreeRTOS. The “eFuse library” implemented eFuse APIs which support eFuse read/write functions. SDK also provides an example CLI command to read/write eFuse.

1.2 Secure eFuse Information

The following table Table 1 eFuse lists the eFuse register definition. The “Index” and “eFuse Length” are the parameters of eFuse API.

Index	eFuse Name	eFuse Register Width (in bit)	eFuse Length (in Byte)	Access Type	Note
1	CUSTK_0	128	16	read/write	Customer key 0
2	CUSTK_1	128	16	read/write	Customer key 1
3	CUSTK_2	128	16	read/write	Customer key 2
4	CUSTK_3	128	16	read/write	Customer key 3
5	RESERVED_0	128	16	read/write	
6	RESERVED_1	128	16	read/write	
7	RESERVED_2	128	16	read/write	
8	RESERVED_3	128	16	read/write	
11	EFUSE_SBC_PUBK_HASH0_0~7	128	16	read/write	SBC Key Hash0
12	EFUSE_SBC_PUBK_HASH0_8~15	128	16	read/write	SBC Key Hash0
13	EFUSE_SBC_PUBK_HASH1_0~7	128	16	read/write	SBC Key Hash1
14	EFUSE_SBC_PUBK_HASH1_8~15	128	16	read/write	SBC Key Hash1
16	EFUSE_SBC_EN	1	1	read/write	Enable Secure Boot
19	EFUSE_SW_JTAG_CON	1	1	read/write	Enable SW JTAG Control (HW CTRL)
22	EFUSE_PL_AR_EN	1	1	read/write	Enable bootloader Anti-Rollback
28	EFUSE_DEVICEEN_DIS	1	1	read/write	
29	EFUSE_DBGEN_DIS	1	1	read/write	disable mcusys JTAG DBGEN signal (disable normal world debug)
30	EFUSE_SPIDEN_DIS	1	1	read/write	disable mcusys JTAG SPIDEN signal (disable secure world debug)
38	EFUSE_PLL_DFT_EN	1	1	read/write	Enable TOP_PLL in BROM
43	EFUSE_USBDL_DIS	1	1	read/write	Disable USB Download feature
44	EFUSE_UARTDL_DIS	1	1	read/write	Disable UART Download feature
49	EFUSE_SBC_PUBK_HASH0_DIS	1	1	read/write	Disable SBC Public key HASH0
50	EFUSE_SBC_PUBK_HASH1_DIS	1	1	read/write	Disable SBC Public key HASH1
58	HUID	128	16	read only	HW Unique ID
72	JTAG_DIS	1	1	read/write	HW JTAG disable
85	EFUSE_PL_VER	128	16	read/write	Preloader version for anti rollback VER0: Byte 0~3; VER1: Byte 4~7 VER2: Byte 8~B; VER3: Byte C~F
86	NS_SW_VER_0	128	16	read/write	RTOS version for anti rollback
87	NS_SW_VER_1	128	16	read/write	RTOS version for anti rollback
88	NS_SW_VER_2	128	16	read/write	RTOS version for anti rollback
89	NS_SW_VER_3	128	16	read/write	RTOS version for anti rollback

90	EFUSE_SBC_ALGO_0	1	1	read/write	Enable SBC algo 0
26	EFUSE_SBC_ALGO_1	1	1	read/write	Enable SBC algo 1

Table 1 eFuse register definition

1.3 Secure eFuse Status Code

If eFuse APIs return a non-zero status code, please refer to the following table for the status return code.

Status Code	
0	eFuse is ok
1	eFuse is busy
2	input parameters is invalid
3	incorrect length of eFuse value
4	can't write a zero eFuse value
5	the eFuse can't be written
6	incorrect eFuse access type
7	the eFuse has been blown
8	the eFuse feature is not implemented
9	check eFuse read failed after eFuse is written
10	token's eFuse value and lenth is not match with that defined in function "eFuse_write"
18	input eFuse value is out of sepcification
19	efuse API fail

Table 2 eFuse Status Code

1.4 Sample Usage

A test program “ewriter” reads/writes eFuse for testing. Followings are the usage:

1.4.1 Hal API

Step1. Include hal_efuse.h header

Step2. How to use ewriter in hal layer

hal_efuse_status_t hal_efuse_ewriter(uint32_t cmd, uint32_t index, uint32_t length, \
uint32_t *buf)

1.4.2 Command line

Step1. Enable ewriter command

{mt7933}/middleware/MTK/minicli/inc/minicli_cmd_table.h

```
diff --git a/inc/minicli_cmd_table.h b/inc/minicli_cmd_table.h
index ea1bbed..9b6f04c 100755
--- a/inc/minicli_cmd_table.h
+++ b/inc/minicli_cmd_table.h
@@ -71,6 +71,7 @@ extern "C" {
 #else
 #define FOTA_CLI_ENTRY
 #endif
+#include "efuse_cli.h"


 #if 0
@@ -453,6 +454,7 @@ extern "C" {
                                         IP_CLI_ENTRY            \
                                        WIFITEST_CLI_ENTRY       \
                                         MINISUPP_CLI_ENTRY      \
+                                        EFUSE_DRV_CLI_ENTRY     \
                                         BSP_DVT_CLI_CMDS


 #else
@@ -517,6 +519,7 @@ extern "C" {
                                         IP_CLI_ENTRY            \
                                        WIFITEST_CLI_ENTRY       \
                                         MINISUPP_CLI_ENTRY      \
+                                        EFUSE_DRV_CLI_ENTRY     \
                                         THERMAL_CLI_ENTRY
 #else
```

Figure 1: minicli_cmd_table.h

Step2. How to use ewriter

efusedrv ewriter_index <read/write> <index> <length> <data byte0~3(writing only)> <data byte4~7(writing only)> <data byte8~11(writing only)> <data byte12~15(writing only)>

read index 16 >> efusedrv ewriter_index 0 16 1

write index 16 >> efusedrv ewriter_index 1 16 1 0x1

read index 1 >> efusedrv ewriter_index 0 1 16

write index 1 >> efusedrv ewriter_index 1 1 16 0x778899aa 0xaabbccef 0xabcdefef 0x12345678

2 MT793X BROM ECDSA-p256, p384, p521 verification support

2.1 Bootloader verification

Figure 2 show the bootloader verification process in BootROM. To enable the secure boot, blow EFUSE_SBC_EN. If the eFuse bit is not blown, BootROM skips all verification steps.

First, BootROM reads EFUSE_SBC_ALGO_MASK0 and EFUSE_SBC_ALGO_MASK1 to check the verification algorithm. Note that BootROM uses ECDSA-p384 or ECDSA-p521 to verify bootloader only if EFUSE_SBC_ALGO_MASK1 is blown.

Next, BootROM parses the public key from the public TLV of the bootloader. If the type of the TLV and the verification algorithm are matched, then the BootROM verifies the validation of the key. Note that to be compatible with the ECO3 chip, the bootloader may append multiple kinds of public keys to the TLV. ECO3 chip only identifies the IMAGE_TLV_PUB_KEY_EC256(0x60) and skips the other. For the ECO4 chip, as described above, BootROM skips all TLV types except the one that matches the verification algorithm.

BootROM then verifies the length and the hash of the public key get from the public TLV of bootloader. Note that the max length of the key is EC521(158 bytes), the public key larger than 158 bytes is invalid. Due to the size of the hash of the public key, the eFuse field (total 512 bits) only blows one hash in EC521 key (512 bits) and EC384 key (384 bits) (start from EFUSE_SBC_PUBK_HASH0 to EFUSE_SBC_PUBK_HASH1) and blow up to 2 hashes in EC256 key (256 bits).

If the hash verification of the key is successful, BootROM parses the signature from the bootloader image. Like the process of parsing public key, BootROM only verifies the signature that matches the verification algorithm. Note that to be compatible with the ECO3 chip, the bootloader may append multiple kinds of signatures to the TLV. ECO3 chip only identifies the IMAGE_TLV_ECDSA256(0x22) and skips the other. For the ECO4 chip, as described above, BootROM skips all TLV types except the one that matches the verification algorithm.

BootROM then calculates the hash of the bootloader image and uses the hash, public key, and signature to verify the validation of the bootloader image.

Note: To simply BootROM verification process in the ECO4 chip, flash tool must guarantee the sequence of the TLV if the bootloader appends multiple kinds of key hashes and signatures. The TLV appending must follow the sequence IMAGE_TLV_PUB_KEY_EC256, IMAGE_TLV_ECDSA256 -> IMAGE_TLV_PUB_KEY_EC384, IMAGE_TLV_ECDSA384 -> IMAGE_TLV_PUB_KEY_EC521, IMAGE_TLV_ ECDSA521.

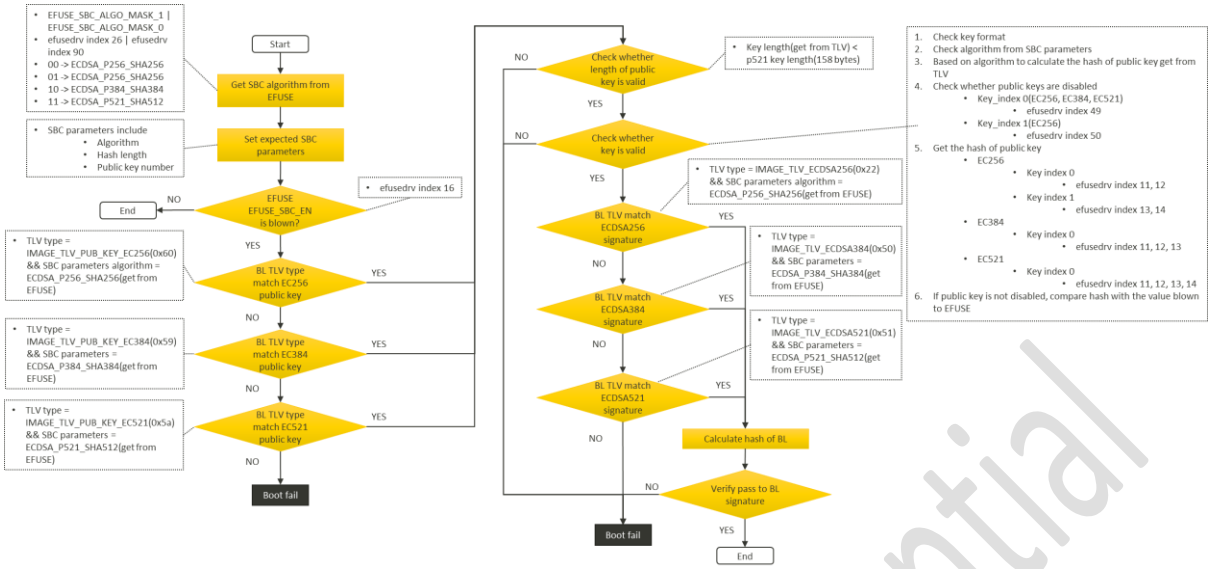

Figure 2: Bootloader verification process

Exhibit 1 Terms and Conditions

Your access to and use of this document and the information contained herein (collectively this "Document") is subject to your (including the corporation or other legal entity you represent, collectively "You") acceptance of the terms and conditions set forth below ("T&C"). By using, accessing or downloading this Document, You are accepting the T&C and agree to be bound by the T&C. If You don't agree to the T&C, You may not use this Document and shall immediately destroy any copy thereof.

This Document contains information that is confidential and proprietary to MediaTek Inc. and/or its affiliates (collectively "MediaTek") or its licensors and is provided solely for Your internal use with MediaTek's chipset(s) described in this Document and shall not be used for any other purposes (including but not limited to identifying or providing evidence to support any potential patent infringement claim against MediaTek or any of MediaTek's suppliers and/or direct or indirect customers). Unauthorized use or disclosure of the information contained herein is prohibited. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for Your unauthorized use or disclosure of this Document, in whole or in part.

MediaTek and its licensors retain titles and all ownership rights in and to this Document and no license (express or implied, by estoppels or otherwise) to any intellectual propriety rights is granted hereunder. This Document is subject to change without further notification. MediaTek does not assume any responsibility arising out of or in connection with any use of, or reliance on, this Document, and specifically disclaims any and all liability, including, without limitation, consequential or incidental damages.

THIS DOCUMENT AND ANY OTHER MATERIALS OR TECHNICAL SUPPORT PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT, IF ANY, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, COMPLETENESS OR ACCURACY AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MEDIATEK SHALL NOT BE RESPONSIBLE FOR ANY MEDIATEK DELIVERABLES MADE TO MEET YOUR SPECIFICATIONS OR TO CONFORM TO A PARTICULAR STANDARD OR OPEN FORUM.

Without limiting the generality of the foregoing, MediaTek makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MediaTek assume any liability arising out of the application or use of any product, circuit or software. You agree that You are solely responsible for the designing, validating and testing Your product incorporating MediaTek's product and ensure such product meets applicable standards and any safety, security or other requirements.

The above T&C and all acts in connection with the T&C or this Document shall be governed, construed and interpreted in accordance with the laws of Taiwan, without giving effect to the principles of conflicts of law.