Secure boot Hands-on

Fill efuse name and Read secure boot check

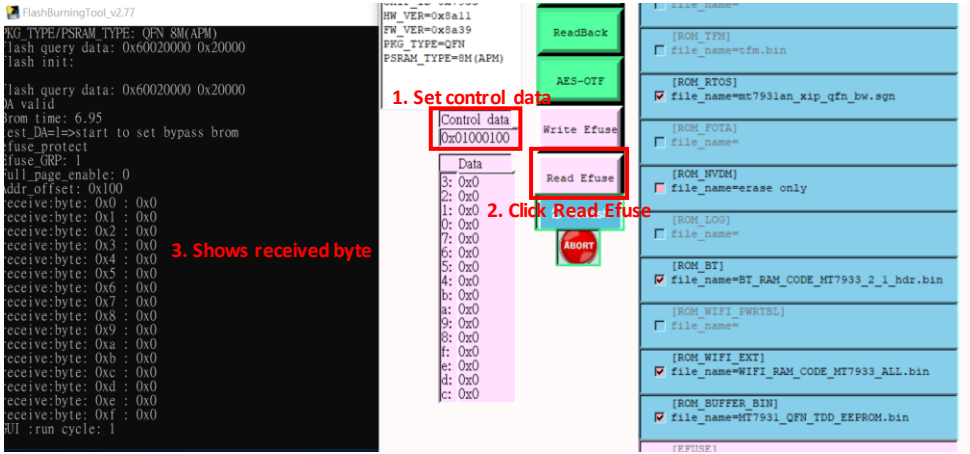
1. CM33 log will notify user that the secure boot verify RTOS fail if not enable secure boot enable bit **EFUSE_SBC_EN** and Fill hash of key in efuse.

```
loader init
  0
Your choose c
secure boot: checking
invalid key 0x3c3c
secure boot: rtos verify fail, status 7 addr 0x18044000 size 0x208000
secure boot: permissive, ignore
jump pc 0x180a0c4d, sp 0x111000
hal_psram_init
```

```
[EFUSE]
enable=n
start_addr=
partition_size=
file_name=test.bin
readback=n
```

2. Fill in Efuse file name in scatter file, which path is as below:
{Top}/out/mt7933_hdk/XXX_scatter.ini

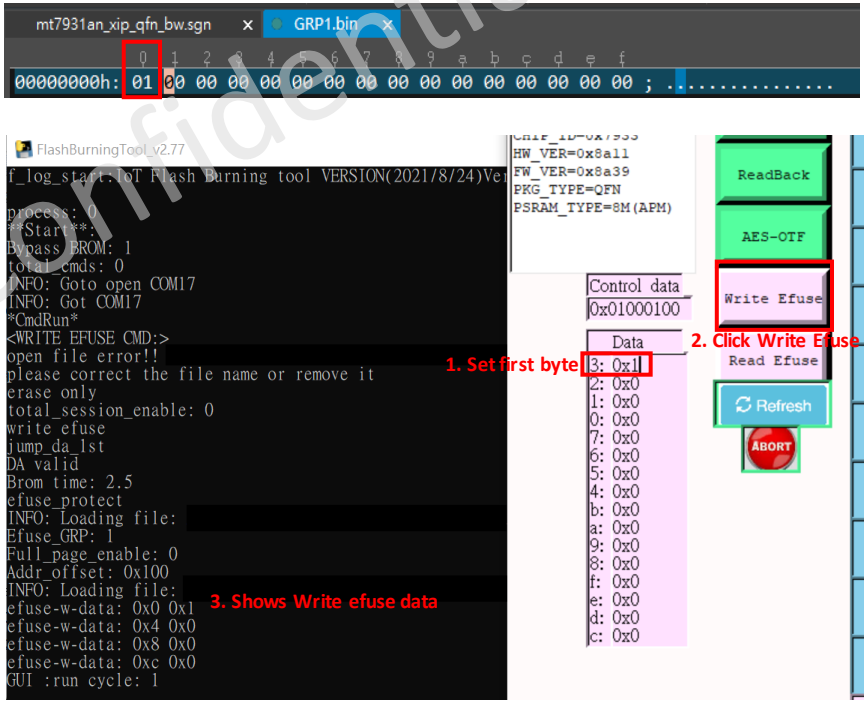
3. Read secure boot check
   - **Set Control data**
   - **Index: 0x01000100**



Enable secure boot check and Generate key pair

4. Enable secure boot check
   - **Set EFUSE_SBC_EN 1 as below**
   - **Control data : 0x01000100**
   - **Address: 0x00**
   - **Index: 0x01**

5. Use OpenSSL to generate an ECC key pair
   - The command generate a file, called *my.pem*, with an ECC key pair (private and public keys) in it.
   - openssl ecparam -genkey -name secp256r1 -out my.pem



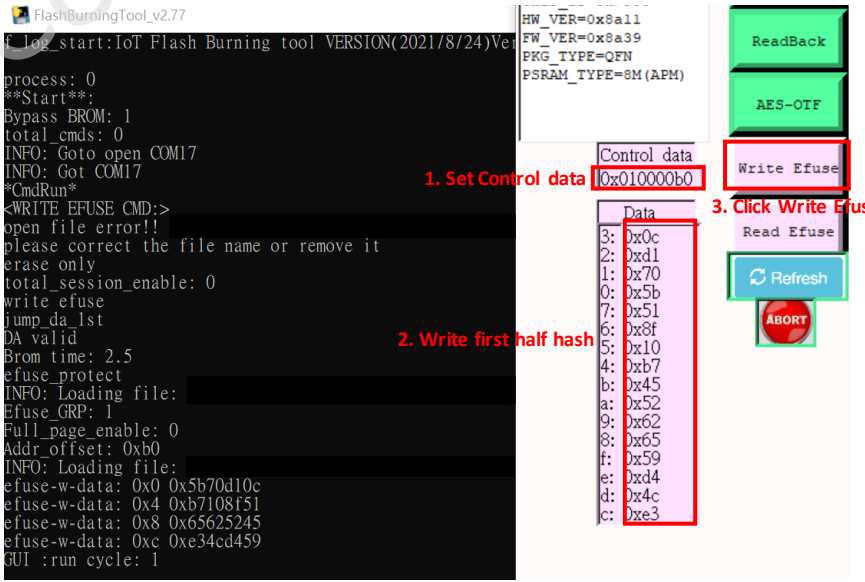
Replace pem file and Generate Hash

6. Replace the mtk-dev.pem and build project
   - {Top}/project/mt7933_hdk/apps/Bootloader/mtk-dev.pem
7. Generate the Hash of the public Key in mtk-dev.pem
   - openssl pkey -inform PEM -in my.pem -pubout -outform DER | sha256sum
8. Blow efuse in the MT793x with the Hash
   - Set control data:0x010000b0
   - Index: first half of hash
   - For example:

0x0c	0x45
0xd1	0x52
0x70	0x62
0x5b	0x65
0x51	0x59
0x8f	0xd4
0x10	0x4c
0xb7	0xe3

Use the following OpenSSL commands to calculate the SHA-256 hash of the public key derived from the private key file my.pem.

```
$ openssl pkey -inform PEM -in my.pem -pubout -outform DER | sha256sum
0cd1705b518f10b7455262659d44ce3132552299dee6f81d3d541fcf544416e -
```



Blow eFuse in the MT793X with the Hash

9. Blow efuse in the MT793x with the Hash

- Set control data:0x010000c0
- Index: the rest of hash

0x13	0xd3
0x25	0xd5
0x52	0x41
0x29	0xfc
0x9d	0xf5
0xee	0x44
0x6f	0x41
0x81	0x6e

10. Check bootloader verify RTOS procedure.

```
loader init
  0
Your choose c
secure boot: checking
jump pc 0x180a0c4d, sp 0x111000
hal_psram_init
```

Use the following OpenSSL commands to calculate the SHA-256 hash of the public key derived from the private key file my.pem.

```
$ openssl pkey -inform PEM -in my.pem -pubout -outform DER | sha256sum
0cd1705b518f10b74552626559d44ce3132552299dee6f81d3d541fcf544416e -
```