CV181x/CV180x eFuse User Guide

Version： 0.4

Release date: 2023-02-06

CONTENTS

Revision History

Revision	Date	Description
0.1	2022-06-01	Initial
0.2	2022-09-28	Rename chip
0.3	2023-02-01	Update the secure boot efuse burning process
0.4	2023-02-06	CV181x/CV180x Document Fusion

CHAPTER

ONE

DISCLAIMER

Terms and Conditions

The document and all information contained herein remain the CVITEK Co., Ltd' s ("CVITEK") confidential information, and should not disclose to any third party or use it in any way without CVITEK' s prior written consent.

User shall be liable for any damage and loss caused by unauthority use and disclosure.

CVITEK reserves the right to make changes to information contained in this document at any time and without notice.

All information contained herein is provided in "AS IS" basis, without warranties of any kind, expressed or implied, including without limitation mercantability, non-infringement and fitness for a particular purpose.

In no event shall CVITEK be liable for any third party' s software provided herein, User shall only seek remedy against such third party.

CVITEK especially claims that CVITEK shall have no liable for CVITEK' s work result based on Customer' s specification or published shandard.

Contact Us

Address Building 1, Yard 9, FengHao East Road, Haidian District, Beijing, 100094, China

Building T10, UpperCoast Park, Huizhanwan, Zhancheng Community, Fuhai Street, Baoan District, Shenzhen, 518100, China

Phone +86-10-57590723 +86-10-57590724

Website https://www.sophgo.com/

Forum https://developer.sophgo.com/forum/index.html

CHAPTER

TWO

EFUSE USER GUIDE

2.1 eFuse Overview

The chip integrates eFuse space, which can be used for secure boot and user-defined area of 448 bits.

Please refer to *eFuse user writable area* and *eFuse security setting fields* for specific eFuse partitions.

Table 2.1: eFuse user writable area

Name	Size	Comment
USER	40 Bytes	User-defined areas
DEVICE_ID	8 Bytes	Serial number of device
HASH0_PUBLIC	32 Bytes	RSA public key hash value for secure boot
LOADER_EK	16 Bytes	AES encryption key for secure boot
DEVICE_EK	16 Bytes	User-defined areas can be locked
SECUREBOOT	4 Bytes	Enable secure boot

Table 2.2: eFuse security setting fields

Name	Comment
LOCK_HASH0_PUBLIC	Lock HASH0_PUBLIC, making this area unreadable
LOCK_LOADER_EK	Lock LOADER_EK, making this area unreadable
LOCK_DEVICE_EK	Lock DEVICE_EK so that this area cannot be read or written
SECUREBOOT	Enable secure boot

2.2 Secure Boot eFuse Setup Process

Attention: eFuse cannot be erased after writing 1 every bit (only allowed to change from 0 to 1), please pay attention before writing. After the specified eFuse is locked, it can no longer be read or written. Please pay attention before locking.

Cvitek provides u-boot command and Linux library to access eFuse. The following process uses u-boot command as an example.

2.2.1 Viewing the Content of Key

To view the key content on a PC:

```
# View AES keys
host$ xxd -p -c 256 loader_ek.key
668f8b6655a89f7cb8ee5cbd6f2c914e

# Obtain RSA public key sha256 value required for signature verification
# When executing the signing script fipsign.py, the script will print the required␣
↪sha256 value, as follows:
host$ ./fipsign.py ......
Host$ .......
Host$ INFO:root:KPUB_
↪HASH:978bc2031b9377dadb4c7c34467ee985806a63a3ac8ee293a3f0eddcd2b789d8
Host$ .......
```

- KPUB_HASH: The following string is the required sha256 value

2.2.2 Writing a Key

1. Write loader_ek.key into the "encryption key" area of eFuse, the data is an array of 16, expressed as 32 numbers in hexadecimal. Skip this step if encryption is not used.

```
u-boot# efusew LOADER_EK 668f8b6655a89f7cb8ee5cbd6f2c914e
```

2. Write the sha256 value required for signature verification into the "SHA256 summary required for signature verification" area of eFuse. The data is an array of 32, expressed as 64 numbers in hexadecimal.

```
u-boot# efusew LOADER_EK 668f8b6655a89f7cb8ee5cbd6f2c914e
```

3. Lock the key area to prevent reading and writing.

```
u-boot# efusew LOCK_LOADER_EK 01
u-boot# efusew LOCK_HASH0_PUBLIC 01
```

2.2.3 Enable secure Boot

1. Enable RSA verification process

```
u-boot# efusew SECUREBOOT 01
```

2. Enable RSA verification and AES decryption process

```
u-boot# efusew SECUREBOOT 02
```

Attention: After the secure boot is enabled, it cannot be changed. Please note that the FIP image has been signed/encrypted before burning.

2.3 eFuse U-boot Command Reference

The u-boot provides the following commands to access eFuse.

- efuser: Dump the eFuse area.
- efusew: Write to eFuse area.

2.3.1 efuser

【Description】 Dump the eFuse area.

【Syntax】 efuser EFUSE_AREA

【Parameter】

Parameter	Description
EFUSE_AREA	eFuse area name, please refer to *eFuse user writable area* and *eFuse security setting fields*.

【Example】 Print user-defined area data

```
cv181x/cv180x# efuser USER
00000000: 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ..........
↪.....
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ..........
↪.....
00000020: 00 00 00 00 00 00 00 00                            ........
cv181x/cv180x#
```

2.3.2 efusew

【Description】 Write to eFuse area

【Syntax】 efuser EFUSE_AREA DATA

【Parameter】

Parameter	Description
EFUSE_AREA	eFuse area name, please refer to *eFuse user writable area* and *eFuse security setting fields*.
DATA	Data for writing into eFuse, expressed in hexadecimal.

【Example】 Write data 030201 to user-defined area

```
cv181x/cv180x# efusew USER 030201
Write eFuse USER(0) with:
00000000: 03 02 01                                           ...
cv181x/cv180x# efuser USER
00000000: 03 02 01 00 00 00 00 00 00 00 00 00 00 00 00 00    ..........
↪.....
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ..........
↪.....
00000020: 00 00 00 00 00 00 00 00                            ........
cv181x/cv180x#
```

2.4 eFuse API Reference

The eFuse API is located in the CIPHER module, providing the following APIs:

- CVI_EFUSE_GetSize: Query the size of the eFuse area.
- CVI_EFUSE_Read: Read eFuse area.
- CVI_EFUSE_Write: Write to eFuse area.
- CVI_EFUSE_EnableSecureBoot: Enable secure boot.
- CVI_EFUSE_IsSecureBootEnabled: Query the secure boot status.
- CVI_EFUSE_EnableFastBoot: Enable fast boot.
- CVI_EFUSE_IsFastBootEnabled: Query the fast boot status.
- CVI_EFUSE_Lock: Lock the eFuse area.
- CVI_EFUSE_IsLocked: Query whether the eFuse area is locked.

2.4.1 CVI_EFUSE_GetSize

【Description】

Query the size of the eFuse area.

【Syntax】

```
CVI_S32 CVI_EFUSE_GetSize(CVI_EFUSE_AREA_E area, CVI_U32 *size);
```

【Parameter】

Parameter	Description	Input/Output
area	Specify eFuse area	Input
size	eFuse area size (unit: byte)	Output

【Return Value】

Return Value	Description
>= 0	Success
< 0	Failure, see error code.

【Requirement】

- Header files: cvi_type.h cvi_unf_cipher.h
- Library files: libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.2 CVI_EFUSE_Read

【Description】

Read the eFuse area.

【Syntax】

```
CVI_S32 CVI_EFUSE_Read(CVI_EFUSE_AREA_E area, CVI_U8 *buf, CVI_U32 buf_size);
```

【Parameter】

Parameter	Description	Input/Output
area	Specify eFuse area	Input
buf	Used to store eFuse data	Output
buf_size	Buffer size (unit: byte)	Input

【Return Value】

Return Value	Description
>= 0	Success
< 0	Failure, see error code.

【Requirement】

- Header files: cvi_type.h cvi_unf_cipher.h
- Library files: libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.3 CVI_EFUSE_Write

【Description】

Write to eFuse area.

【Syntax】

```
CVI_S32 CVI_EFUSE_Write(CVI_EFUSE_AREA_E area, const CVI_U8 *buf, CVI_U32 buf_size);
```

【Parameter】

Parameter	Description	Input/Output
area	Specify eFuse area	Input
buf	Data to write to eFuse	Input
buf_size	Buffer size (unit: byte)	Input

【Return Value】

Return Value	Description
>= 0	Success
< 0	Failure, see error code.

【Requirement】

- Header files: cvi_type.h cvi_unf_cipher.h

- Library files：libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.4 CVI_EFUSE_EnableSecureBoot

【Description】

Enable secure boot.

【Syntax】

```
CVI_S32 CVI_EFUSE_EnableSecureBoot(void);
```

【Parameter】

None.

【Return Value】

Return Value	Description
>= 0	Success
< 0	Failure, see error code.

【Requirement】

- Header files：cvi_type.h cvi_unf_cipher.h
- Library files：libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.5 CVI_EFUSE_IsSecureBootEnabled

【Description】

Query the secure boot status.

【Syntax】

```
CVI_S32 CVI_EFUSE_IsSecureBootEnabled(void);
```

【Parameter】

None.

【Return Value】

Return Value	Description
> 0	Secure boot is enabled
0	Secure boot is not enabled
< 0	Failure, see error code.

【Requirement】

- Header files：cvi_type.h cvi_unf_cipher.h
- Library files：libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.6 CVI_EFUSE_EnableFastBoot

【Description】

Enable fast boot.

【Syntax】

```
CVI_S32 CVI_EFUSE_EnableFastBoot(void);
```

【Parameter】

None.

【Return Value】

Return Value	Description
0	Fast boot enabled
< 0	Failure, see error code.

【Requirement】

- Header files: cvi_type.h cvi_unf_cipher.h
- Library files: libsys.a

【Note】 None.

【Example】 Refer to `sample_fastboot.c`。

> **Attention:** Cannot be changed after the fast boot is enabled.

2.4.7 CVI_EFUSE_IsFastBootEnabled

【Description】

Query the fast boot status.

【Syntax】

```
CVI_S32 CVI_EFUSE_IsFastBootEnabled(void);
```

【Parameter】

None.

【Return Value】

Return Value	Description
0	Fast boot enabled
< 0	Fast boot not enabled

【Requirement】

- Header files: cvi_type.h cvi_unf_cipher.h
- Library files: libsys.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.8 CVI_EFUSE_Lock

【Description】

Lock the eFuse area.

【Syntax】

```
CVI_S32 CVI_EFUSE_Lock(CVI_EFUSE_LOCK_E lock);
```

【Parameter】

Parameter	Description	Input/Output
area	Specify the eFuse area to lock	Input

【Return Value】

Return Value	Description
>= 0	The specified eFuse partition is locked.
< 0	Failure, see error code.

【Requirement】

- Header files：cvi_type.h cvi_unf_cipher.h
- Library files：libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.4.9 CVI_EFUSE_IsLocked

【Description】

Query whether the eFuse area is locked.

【Syntax】

```
CVI_S32 CVI_EFUSE_IsLocked(CVI_EFUSE_LOCK_E lock);
```

【Parameter】

Parameter	Description	Input/Output
area	Specify the eFuse area to lock	Input

【Return Value】

Return Value	Description
> 0	The specified eFuse partition is locked.
0	The specified eFuse partition has not been locked.
< 0	Failure, see error code.

【Requirement】

- Header files：cvi_type.h cvi_unf_cipher.h
- Library files：libcipher.a

【Note】 None.

【Example】 Refer to `sample_efuse.c`。

2.5 Data Types

The relevant data types and data structures are defined as follow:

- CVI_EFUSE_AREA_E: Define eFuse area.
- CVI_EFUSE_LOCK_E: Define the lock corresponding to eFuse area.

2.5.1 CVI_EFUSE_AREA_E

【Description】

Define eFuse area

【Definition】

```
typedef enum {
    CVI_EFUSE_AREA_USER,
    CVI_EFUSE_AREA_DEVICE_ID,
    CVI_EFUSE_AREA_HASH0_PUBLIC,
    CVI_EFUSE_AREA_LOADER_EK,
    CVI_EFUSE_AREA_DEVICE_EK,
    CVI_EFUSE_AREA_LAST
} CVI_EFUSE_AREA_E;
```

【Member】

Member	Description
CVI_EFUSE_AREA_USER	User defined area
CVI_EFUSE_AREA_DEVICE_ID	Device serial number area
CVI_EFUSE_AREA_HASH0_PUBLIC	Secureboot RSA public key hash value area
CVI_EFUSE_AREA_LOADER_EK	Secureboot AES encryption key area
CVI_EFUSE_AREA_DEVICE_EK	DEVICE_EK area
CVI_EFUSE_AREA_LAST	End identification

【Note】

None.

【Related Data Type and Interface】

CVI_EFUSE_GetSize, CVI_EFUSE_Read, CVI_EFUSE_Write

2.5.2 CVI_EFUSE_LOCK_E

【Description】

Define the lock corresponding to eFuse area.

【Definition】

```
typedef enum {
    CVI_EFUSE_LOCK_HASH0_PUBLIC,
    CVI_EFUSE_LOCK_LOADER_EK,
    CVI_EFUSE_LOCK_DEVICE_EK,
    CVI_EFUSE_LOCK_LAST
} CVI_EFUSE_LOCK_E;
```

【Member】

Member	Description
CVI_EFUSE_LOCK_HASH0_PUBLIC	Lock secureboot RSA public key hash value area
CVI_EFUSE_LOCK_LOADER_EK	Lock secureboot AES encryption key area
CVI_EFUSE_LOCK_DEVICE_EK	Lock DEVICE_EK area
CVI_EFUSE_LOCK_LAST	End identification

【Note】

None.

【Related Data Type and Interface】

CVI_EFUSE_Lock, CVI_EFUSE_IsLocked