NT96680 Crypto via key manager Application Note

Release date: 2019/02/27

Table of Content

Revision History

Date	Contents
2019/02/27	1. Add get secure enable or not / key field set or not API @ u-boot
2018/12/24	1. Add Loader binary file layout 说明
2018/12/19	1st release of secure boot

1. Crypto engine with OTP key to encrypt / decrypt binary file application

1.1. 芯片安全启动使用指南

文字定义:

- Cypher text loader : 表示加密过后的 loader
- Plan text loader : 表示未加密过后的 loader
- 透过 OTP/efuse 直送 key 到 Encrypt/ decrypt engine : 统称透过 key manager 来加解密

需求描述	指引
Cypher text(密文)loader vs plan text(明文)loader 镜像格式介绍	
Boot rom 启动(cypher) loader 过程介绍	
Cypher text(密文)loader 生成介绍	
OTP/efuse 中烧写密钥/禁止 JTAG 方法提供与介绍	
OTP/efuse 开启安全导引功能方法介绍	

1.2. 加解密 API 在开机流程的角色

(1).Boot Rom (解密)→ Loader

(2).Loader(解密) → u-boot

(3).u-boot(解密) → linux

(4). Linux 透过 crypto framework 自行加解密 Bin 檔

请参考下图 1-2-1

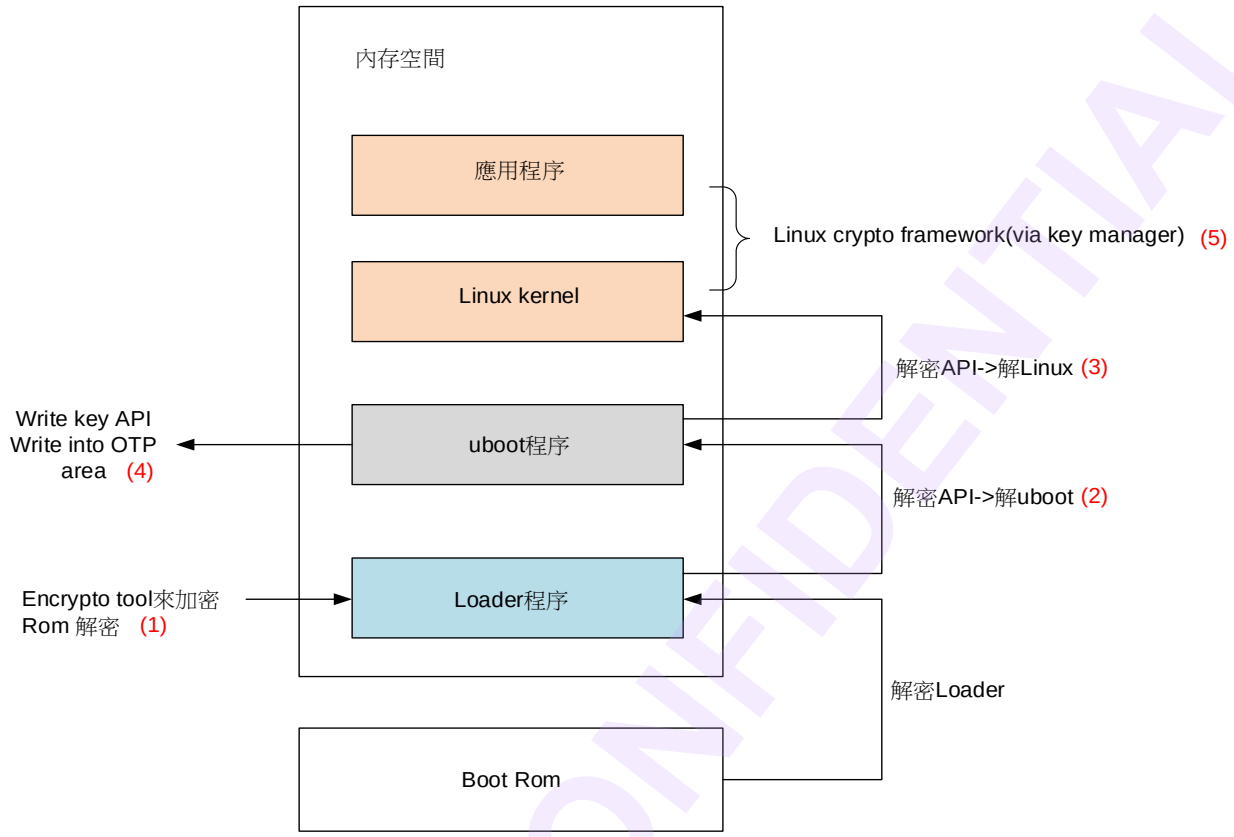


Figure 1-2-1

1.3. 我们会提供

(1). Offline encrypt tool 提供制作出 cypher Loader(1)

(2). 解密 API(@loader) (2)

(3). 解密 API(@uboot) (3)

(4). Write key API & Secure enable API & JTAG disable API (@uboot) (4)

(5). Crypto framework by using SoC key manager (@Linux) (5)

(6). Secure boot enable 激活 secure boot (only available in u-boot phase) (4)

(7). JTAG disable Disable JTAG (only available in u-boot phase) (4)

- Supports four key area for use (AES128)①

①Secure boot use 1st key set field only.

1.4. Hardware PCB layout guide (硬件上需上的组件)

- 若要对 E-Fuse 做写入的动作，NC96685/NC98535 J7 这一 Pin 需要供 1.8V
- 开机时序:0.9V 先开，再开 1.8V(E-Fuse 只有用到这两组电源)
- 关机时序:1.8 先关，再关 0.9V

AD24	AVCC_USB	P	Analog 3.3V power for USB interface
AE24, AF24	AGND_USB	P	Ground for USB
J7	TP	-	Connected 100K Ohm resistor to ground

1.5. Firmware function guide

1.5.1. Loader with

.1.5.1.1. Cypher text loader generate step by step

- General Description

NT9668x support a feature called secured boot. When this feature is enabled, NT9668x will boot by an cypher text loader. The "cypher text loader" is generated by set SECUREBOOT = Secure. For detail descriptions of secure boot, please contact Novatek AE.

- 操作步骤

Following example will enable generating secure boot loader:

(1). 请在 ModelConfig_xxx.txt 把 SECUREBOOT = Secure 设起来 (方式如下)

```
# Example for your ModelConfig_XXX.txt

SECUREBOOT = Secure
```

```
# [SECUREBOOT] : generate loader for secure boot
# Normal
# Secure
SECUREBOOT = Secure
```

(2). 输入 Key, 位置在 Loader_code_base\Tools\Bin\aes.txt 内容如下 (②)

假设你要输入 key 是 0102030405060708091011121314151 6 请照底下顺序输入

#后面是不使用

aes.txt

```
1  #0102030405060708091011121314151 6
2  0102030405060708091011121314151 6
3  #1314151609101112050607080102030 4
```

到 MakeCommon folder 去 make binary 檔, 结果会出现一些 Key/hash 相关讯息, 如下

②: 这里是有可能需要贵司移植的地方, 目前我们 key 是放在文本文件里面, 然后提供了顺序说明

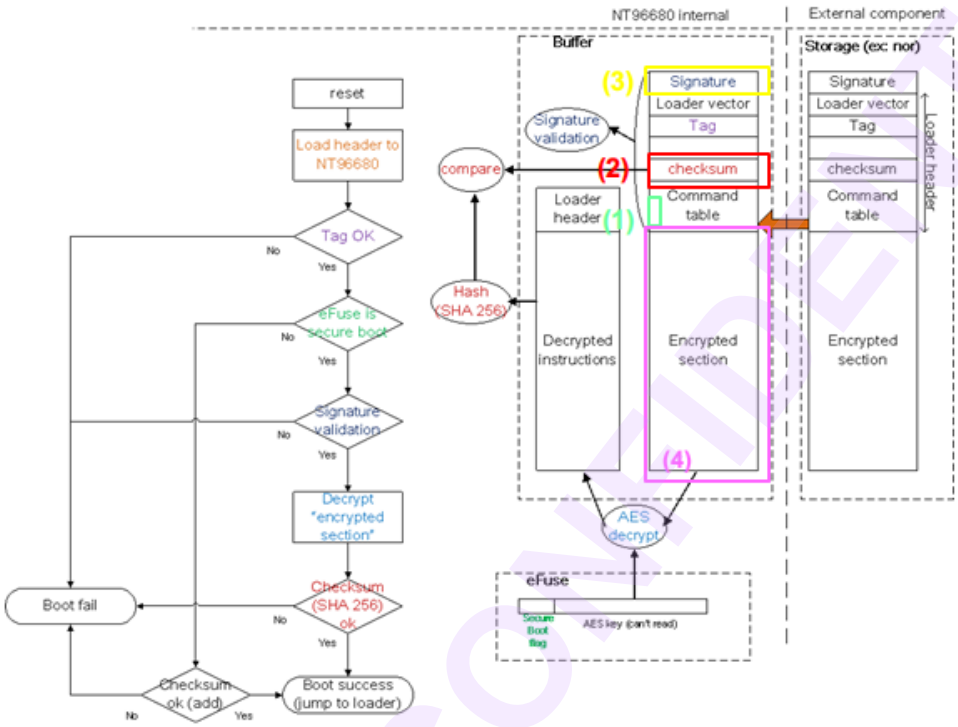
```
checkSumValue=0xB3A6
uiNewCheckSum = 0x7FD9Writing 'Loader98535_Data/Release/LD98535A.bin' success
Writing size 0x8000 success...
encrypt boot version : 2018-0321-1.0006
chip version [NT96680]
secure boot: 1          表示secure enable
Reading 'Loader98535_Data/Release/LD98535A.bin'...
file size = 32768

config ram count 0x4

==Display binary file info== Begin!!!
VersionNumber: 04,00,00,00
==Display binary file info== End

total_bin_size after padding:32768, Dram param number:4

##ui32BinDataSumValue: 0x2147B19F
CheckSum: 0x4E61

AES-Key:
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
swap =>
13 14 15 16 09 10 11 12 05 06 07 08 01 02 03 04
##ui32BinDataSumValue: 0x274C338A
Data CheckSum: 0xCC76

##ui32BinDataSumValue: 0x2148CC76
Re-CheckSum: 0x338A

SHA256: 0xdcf2042e 0xd98bd26b 0x152eb033 0x6e1d233e
        0x4d646eef 0x4e88604 0xfbbebd0 0x8d1e0a88
Sig: 0x27136465 0x5329c669 0xa04fef75 0x6f21d16e
        0xab9f16fd 0xb8421bfa 0x4dce19e4 0xca44e644
LINUX system !!
AES-Key:
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
swap =>
13 14 15 16 09 10 11 12 05 06 07 08 01 02 03 04
Encypt Sig: 0xcf745676 0xcfff8fc 0xc2387369 0x6c9a50d7
        0x841405cc 0xd7963584 0x710fb9aa 0x2738c398
Writing 'Loader98535_Data/Release/LD98535A.bin'...
write size = 32768
Save BIN OK.
All OK.
```

这样 cypher text loader 就产生了 ➔ 请注意要配合后面提到的 secure enable 后才能使用此一 loader

PS, 可以参考整份文件会 reference 到的第三方网站,key order 都对齐它

.1.5.1.2. Cypher text loader 鏡像格式介紹

- Loader image layout



Loader透过offline tool 产生flow (1)→(2)→(3)→(4)

(1).针对32K loader算checksum(原本flow)
此时(2) (3)字段都是0

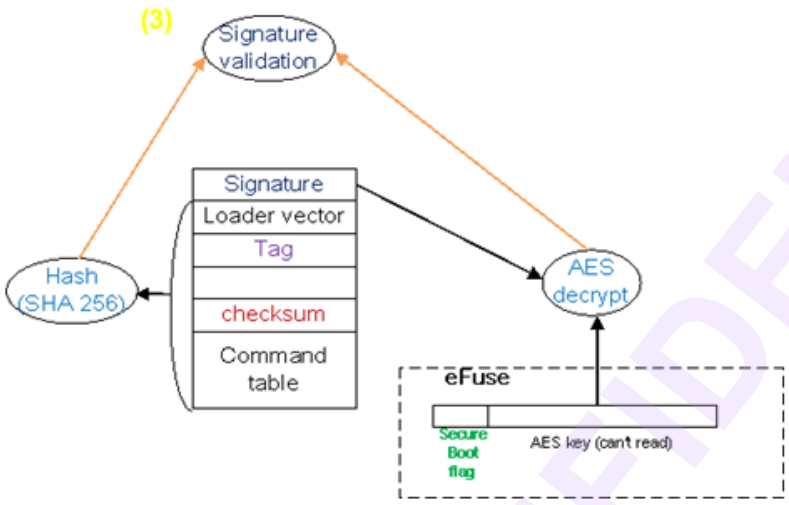
(2).针对32K loader算sha256 hash 值写回(2)字段
此时(3)字段是0

(3).針對512B header 算sha256 hash 用 (aes-cbc-128 bit) encrypt 當成signature
寫回(3)欄位

(4).From 0x200~0x8000 → aes-cbc-128 bit 整片 encrypt成cypher-text

Rom code解回loader flow(4)→(3)→(2)→(1)

- 其中第三點"signature"簽章, 我們作法是
  把 Header 用 hash(SHA256)成一個小長度 size, 再透過 OTP 內的 key 用 AES-CBC-128 加密成密文放在簽章欄位



.1.5.1.3. Decrypt API (Encrypt)

[Description]

Decrypt (Encrypt) specific data from DRAM via OTP key③

③Loader 有 code size 限制, 尽量避免在 loader 调用太多 API, 以免 size 超过限制

[Function]

Function	Description
UINT32 crypto_data_operation(EFUSE_OTP_KEY_SET_FIELD key_set, CRYPT_OP crypt_op_param)	Encrypt or decrypt via OTP key

[Parameter]

Parameter	Description
EFUSE_WRITE_KEY_SET_TO_OTP_FIELD	2nd / 3rd / 4th key set (2~4 key set are available)
CRYPT_OP	Structure including *CRYPTO_OPMODE op_mode; ///< Operation Mode (now support ECB only)* *CRYPTO_TYPE en_de_crypt;///< CRYPTO_ENCRYPT or CRYPTO_DECRYPT* *UINT32 src_addr; ///< Source address* *UINT32 dst_addr; ///< Destination address*

	UINT32 length; ///< length

[Return value]

Value	Description
E_OK	Success
Other values	Error

- **Example：**把底下的 key 写进第三组 Key field

(1) Key order. 举例. 如果 Key 是 0x01,0x02,0x03,0x04, 0x05,0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16

请宣告成

static UINT8 key_sample[16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16};

!!!届时从公司内部 server 拿到的 key 也请转成此 order!!!

```
static UINT8 key_sample[16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16};
static UINT8 out_put[16] = {0};
    CRYPT_OP crypt_op_param;
UINT32 index_cnt;
crypt_op_param.op_mode = CRYPTO_EBC;
crypt_op_param.en_de_crypt = CRYPTO_ENCRYPT;
crypt_op_param.src_addr = (UINT32)key_sample;
crypt_op_param.dst_addr = (UINT32)out_put;    //<<<----------(1)
crypt_op_param.length    = 16;
uart_putSystemUARTStr("\r\n");
if (crypto_data_operation(EFUSE_OTP_1ST_KEY_SET_FIELD, crypt_op_param) != 0) {
    uart_putSystemUARTStr("fail\r\n");
} else {
    for (index_cnt = 0; index_cnt < 16; index_cnt++) {
            uart_putSystemUARTStr(Dec2HexStr2Bytes(out_put[index_cnt]));
            uart_putSystemUARTStr(" ");
        }
        uart_putSystemUARTStr("\r\n");
}
```

Loader 加 / 解密 sample code 如下

```
static UINT8 key_sample[16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16};
static UINT8 out_put[16] = {0};

__attribute__((target("thumb2"))) UINT32 bl_mainFlow(void)
{
    UINT32  uiUpdateFileLen = 0;
    UINT32  uiLoaderFunc = 0;
    UINT32  uiLoaderSize = 32 * 1024; // pre-assume 32KB, actual size is parsed from loader
    CRYPT_OP crypt_op_param;
    UINT32  index_cnt;
    //#NT#2013/04/25#Steven Wang -begin
    //#NT#Show Duty calibration log
    //UINT32  uiStorageFlag = (UINT32)&_load_LOADER_CONFIGRAM_FREQ_PARAM_end_base;
#if (_LOADER_DUTY_CALIBRATION_ == ENABLE && _LOADER_DUTY_CALIBRATION_LOG_ == ENABLE)
    UINT32  uiLoaderAddress;
    UINT32  uiLogSramAddress;
#endif
    //#NT#2013/04/25#Steven Wang -end
    unsigned int adjusted_addr = 0;
    unsigned int adjusted_size = 0;
    UINT32 uiFwBaseAddr = SDRAM_Start_FW;       // FW base address
    DRAM_PARTITION *p_dram_partition = NULL;

    // BaseOfStack is initialized at doRemapLZ.s
    UINT32 uiheapBufferAddr             = BaseOfStack + 0x4000;        // reserve 16KB for tmp buffer usage
    UINT32 uiTmpBufferAddr              = uiheapBufferAddr + FAT_HEAP_BUFFER_SIZE;
    UINT32 uiUpdateBootloaderBufAddr    = uiTmpBufferAddr + 0x4000;
    UINT32 uiUpdateMainBinBufAddr       = SDRAM_Start_FW;
    *(UINT32 *)0xF0290000 = '1';
    // UART initial sequence
    uart_openSystemUART();

    //uart_putSystemUARTStr(">>>>>\r\n");
    //crypto_data_operation(EFUSE_OTP_1ST_KEY_SET_FIELD, NULL);
    // rtc reset shutdown timer
    // rtc_resetShutDownTimer();

    crypt_op_param.op_mode = CRYPTO_EBC;
    crypt_op_param.en_de_crypt = CRYPTO_ENCRYPT;
    crypt_op_param.src_addr = (UINT32)key_sample;
    crypt_op_param.dst_addr = (UINT32)out_put;   //<<<-------------(1)
    crypt_op_param.length   = 16;

    uart_putSystemUARTStr("\r\n");

    if (crypto_data_operation(EFUSE_OTP_1ST_KEY_SET_FIELD, crypt_op_param) != 0) {
        //intf("Encrypt operation fail [%d] set\r\n", (int)(key_set + 1));
        uart_putSystemUARTStr("fail\r\n");
    } else {
        for (index_cnt = 0; index_cnt < 16; index_cnt++) {
            uart_putSystemUARTStr(Dec2HexStr2Bytes(out_put[index_cnt]));
            uart_putSystemUARTStr(" ");
        }
        uart_putSystemUARTStr("\r\n");
    }
```

[See also]

crypto.h

1.5.2. u-boot with

.1.5.2.1. Write key operation

[Description]

1. Write specific key into specific key set
2. Only provide in u-boot

Note : Set specific bit represent key set N is programmed.

[Function]

In uboot :

#include <asm/arch/efuse_protected.h>

Function	Description
INT32 efuse_write_key(EFUSE_WRITE_KEY_SET_TO_OTP_FIELD key_set_index, UINT8 *uc_key)	Write key into specific field

[Parameter]

Parameter	Description
EFUSE_WRITE_KEY_SET_TO_OTP_FIELD	1st / 2nd / 3rd / 4th key set
key	16 bytes key

[Note]

1st key set field is for secure boot.

[Return value]

Value	Description
E_OK	Success
Other values	Error

[See also]

efuse_protected.h

- **Example：** 我要把底下的 key 写进第三组 Key field

  (1) Key order. 举例. 如果 Key 是 0x01,0x02,0x03,0x04, 0x05,0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16

  请宣告成

  UINT8 key_3rd_sample [16] = { 0x01,0x02,0x03,0x04, 0x05,0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };

  !!!届时从公司内部 server 拿到的 key 也请转成此 order!!!

  详细的 example 可以参考 uboot 内的 cmd do_nvt_write_key_cmd @ na51000_utils.c

```
if(efuse_write_key(EFUSE_WRITE_3RD_KEY_SET_FIELD, key_3rd_sample) < 0)
{
    DBG_DUMP("Write key fail\r\n");
}
else
{
    DBG_DUMP("Write key success\r\n");
}
```

.1.5.2.2. Encrypt & Decrypt operation by OTP key

[Description]

Encrypt or decrypt specific data from DRAM

[Function]

Function	Description
ER crypto_data_operation(EFUSE_WRITE_KEY_SET_TO_OTP_FIELD key_set, CRYPT_OP crypt_op_param)	*Encrypt or decrypt data*

[Parameter]

Parameter	Description
EFUSE_WRITE_KEY_SET_TO_OTP_FIELD	2nd / 3rd / 4th key set (2~4 key set are available)
CRYPT_OP	Structure including *CRYPTO_OPMODE op_mode; ///< Operation Mode (now support ECB only)* *CRYPTO_TYPE en_de_crypt;///< CRYPTO_ENCRYPT or CRYPTO_DECRYPT*

	UINT32 src_addr; ///< Source address UINT32 dst_addr; ///< Destination address UINT32 length; ///< length

[Note]

1st key set field is for secure boot.

[Return value]

Value	Description
E_OK	Success
Other values	Error

[See also]

efuse_protected.h

crypto.h

- **Example**

  Assume key already program as 1.3.1

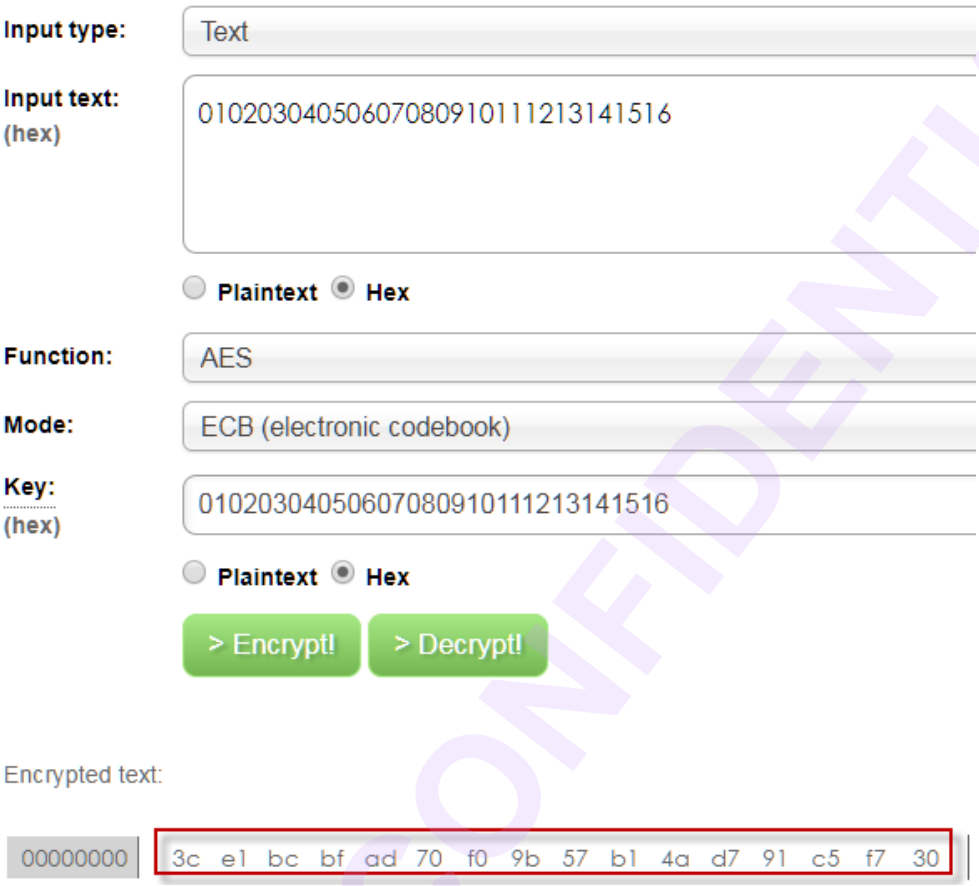
  UINT8 key_2nd [16] = { 0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x10,0x11,0x12,0x13,0x14,0x15,0x16 };

  Input data use same data key_2nd

```
CRYPT_OP        crypt_op_param;
UINT32          index_cnt;
UINT8           ucOutput[16];
UINT8           ucOutput2[16];
crypt_op_param.op_mode = CRYPTO_EBC;
crypt_op_param.en_de_crypt = CRYPTO_ENCRYPT;
crypt_op_param.src_addr = (UINT32)key_2nd;
crypt_op_param.dst_addr = (UINT32)ucOutput;
crypt_op_param.length     = 16;
if(crypto_data_operation(EFUSE_WRITE_2ND_KEY_SET_FIELD, crypt_op_param) != E_OK) {
  emu_msg(("Encrypt operation fail \r\n"));
} else {
  emu_msg(("Encrypt operation success\r\n"));
```

```
            for (index_cnt = 0; index_cnt < 16; index_cnt++) {
                emu_msg(("[%2x]", ucOutput[index_cnt]));
            }
            emu_msg(("\r\n"));
        }
Result
Encrypt operation success – 与下面网站上的结果一致
[3c][e1][bc][bf][ad][70][f0][9b][57][b1][4a][d7][91][c5][f7][30]
        crypt_op_param.op_mode = CRYPTO_EBC;
        crypt_op_param.en_de_crypt = CRYPTO_DECRYPT;
        crypt_op_param.src_addr = (UINT32)ucOutput;
        crypt_op_param.dst_addr = (UINT32)ucOutput2; << 解密回来
        crypt_op_param.length    = 16;

        if(crypto_data_operation(EFUSE_WRITE_2ND_KEY_SET_FIELD, crypt_op_param) != E_OK) {
            emu_msg(("Decrypt operation fail \r\n"));
        } else {
            emu_msg(("Decrypt operation success\r\n"));
            for (index_cnt = 0; index_cnt < 16; index_cnt++) {
                emu_msg(("<--[%2x]", ucOutput2[index_cnt]));
            }
                emu_msg(("\r\n"));
        }
Decrypt operation success
[ 1][ 2][ 3][ 4][ 5][ 6][ 7][ 8][ 9][10][11][12][13][14][15][16] << 还原回原本数据
```

Reference web site Result show as bellow http://aes.online-domain-tools.com/



.1.5.2.3. Encrypt & Decrypt operation by fill crypto engine key field directly
(透过 CPU 填 key 来加解密 → 可以拿来验证 Write key 是否正确)

[Description]

Encrypt or decrypt specific data from DRAM via fill crypto engine

[Function]

Function	Description
ER crypto_data_operation_by_key(UINT8 * key, CRYPT_OP crypt_op_param)	Encrypt or decrypt data

[Parameter]

Parameter	Description
UINT8 *	Key data array if key = 0102030405060708091011121314151 6, the key array will be key_array[16] = {0x01, 0x02, 0x3, 0x4, 0x05, 0x06, 0x7, 0x8, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16};
CRYPT_OP	Structure including CRYPTO_OPMODE op_mode; ///< Operation Mode (now support ECB only) CRYPTO_TYPE en_de_crypt;///< CRYPTO_ENCRYPT or CRYPTO_DECRYPT UINT32 src_addr; ///< Source address UINT32 dst_addr; ///< Destination address UINT32 length; ///< length

[Note]

可以拿这个配合 1.3.1&1.3.2 来验证写进去 Key 是否如预期

[Return value]

Value	Description
E_OK	Success
Other values	Error

[See also]

crypto.h

- **Example**

UINT8 key_2nd [16] = { 0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x10,0x11,0x12,0x13,0x14,0x15,0x16 };

Input data use same data key_2nd

```
CRYPT_OP        crypt_op_param;
UINT32          index_cnt;
UINT8           ucOutput[16];
UINT8           ucOutput2[16];
crypt_op_param.op_mode = CRYPTO_EBC;
crypt_op_param.en_de_crypt = CRYPTO_ENCRYPT;
crypt_op_param.src_addr = (UINT32)key_2nd;
crypt_op_param.dst_addr = (UINT32)ucOutput;
crypt_op_param.length    = 16;
if(crypto_data_operation_by_key(key_2nd, crypt_op_param) != E_OK) {
  emu_msg(("Encrypt operation fail \r\n"));
} else {
  emu_msg(("Encrypt operation success\r\n"));
      for (index_cnt = 0; index_cnt < 16; index_cnt++) {
            emu_msg(("[%2x]", ucOutput[index_cnt]));
   }
  emu_msg(("\r\n"));
}
```

Result

Encrypt operation success – 与下面网站上的结果一致.

[3c][e1][bc][bf][ad][70][f0][9b][57][b1][4a][d7][91][c5][f7][30]

```
crypt_op_param.op_mode = CRYPTO_EBC;
crypt_op_param.en_de_crypt = CRYPTO_DECRYPT;
crypt_op_param.src_addr = (UINT32)ucOutput;
crypt_op_param.dst_addr = (UINT32)ucOutput2; << 解密回来
crypt_op_param.length    = 16;

if(crypto_data_operation(EFUSE_WRITE_2ND_KEY_SET_FIELD, crypt_op_param) != E_OK) {
```

```
        emu_msg(("Decrypt operation fail \r\n"));
    } else {
        emu_msg(("Decrypt operation success\r\n"));
        for (index_cnt = 0; index_cnt < 16; index_cnt++) {
                emu_msg(("<--[%2x]", ucOutput2[index_cnt]));
        }
                emu_msg(("\r\n"));
    }
Decrypt operation success
[ 1][ 2][ 3][ 4][ 5][ 6][ 7][ 8][ 9][10][11][12][13][14][15][16] << 还原回原本数据
```

- **Example : werite key + verify key correct or not flow**

```
static UINT8 key_2nd [16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16};
static UINT8 encrypt_data[16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16};
//与 Server 拿到的 key
//(1). Write 进 OTP
//(2). Encrypt via keymanager
//(3). Encrypt via CPU fill into crypto engine
//比较(2).(3). Result 就可以知道是否烧对
```

详细可参考

.\BSP\u-boot\board\novatek\nvt-na51000\na51000_utils.c → do_nvt_write_key_cmd( )

```
    CRYPT_OP        crypt_op_param;
    UINT32          index_cnt;
    UINT8           Output[16];
    UINT8           Output2[16];
    UINT8           Output3[16];

    UINT32          key_set = EFUSE_OTP_2ND_KEY_SET_FIELD; (依需求看要写进那一组)
    //把 key_2nd 內容写进 OTP ----------------------------(1)
    if (efuse_write_key(key_set, key_2nd) < 0){
        printf("Write [%d] key operation fail\r\n", (int)(key_set + 1));
    } else {
```

```
    printf("Write [%d] key operation success\r\n", (int)(key_set + 1));
}

//Encrypt via OTP by 2nd Key fiele
crypt_op_param.op_mode = CRYPTO_EBC;
crypt_op_param.en_de_crypt = CRYPTO_ENCRYPT;
crypt_op_param.src_addr = (UINT32) encrypt_data;
crypt_op_param.dst_addr = (UINT32)Output;
crypt_op_param.length    = 16;\
//Encrypt via keymanager ----------------------------(2)
if (crypto_data_operation(key_set, crypt_op_param) != 0) {
    printf("Encrypt operation fail [%d] set\r\n", (int)(key_set + 1));
} else {
    printf("Encrypt operation success [%d] set\r\n", (int)(key_set + 1));
    printf("Source =>\r\n");
    for (index_cnt = 0; index_cnt < 16; index_cnt++) {
        printf("%2x ", encrypt_data [index_cnt]);
    }
    printf("\r\n");
    printf("Destination =>\r\n");
    for (index_cnt = 0; index_cnt < 16; index_cnt++) {
        printf("%2x ",Output[index_cnt]);
    }
    printf("\r\n");
}
```

Encrypt operation success – 与下面网站上的结果一致.

[3c][e1][bc][bf][ad][70][f0][9b][57][b1][4a][d7][91][c5][f7][30]

```
crypt_op_param.op_mode = CRYPTO_EBC;
crypt_op_param.en_de_crypt = CRYPTO_DECRYPT;
crypt_op_param.src_addr = (UINT32)Output;
crypt_op_param.dst_addr = (UINT32)Output2; << 解密回来
crypt_op_param.length    = 16;
```

```
    if(crypto_data_operation(key_set, crypt_op_param) != E_OK) {
        emu_msg(("Decrypt operation fail \r\n"));
    } else {
        emu_msg(("Decrypt operation success\r\n"));
        for (index_cnt = 0; index_cnt < 16; index_cnt++) {
                emu_msg(("<--[%2x]", Output2[index_cnt]));
        }
                emu_msg(("\r\n"));
    }
    Decrypt operation success
    [ 1][ 2][ 3][ 4][ 5][ 6][ 7][ 8][ 9][10][11][12][13][14][15][16] << 还原回原本数据

    也可以配合 crypto engine 的 CPU 填 key API 来交叉验证 Write 进 OTP 内的 key 是否正确.
    crypt_op_param.op_mode = CRYPTO_EBC;
    crypt_op_param.en_de_crypt = CRYPTO_ENCRYPT;
    crypt_op_param.src_addr = (UINT32) encrypt_data;
    crypt_op_param.dst_addr = (UINT32)Output3;
    crypt_op_param.length    = 16;
    ///(3). Encrypt via CPU fill into crypto engine
    crypto_data_operation_by_key(key_2nd, crypt_op_param); (走 CPU fill 这路)



    printf("Verification via CPU fill key=>\r\n");
    for (index_cnt = 0; index_cnt < 16; index_cnt++) {
        printf("%2x ", Output3[index_cnt]);
    }
    printf("\r\n");


if (memcmp((void *)Output3, (void *)Output, 16) != 0) {
        printf("write key fail [%d] set\r\n", (int)(key_set + 1));
        printf("OTP operation\r\n");
        for (index_cnt = 0; index_cnt < 16; index_cnt++) {
```

```
                printf("%2x ", Output[index_cnt]);
        }

        printf("\r\n");
        printf("Fill key operation\r\n");
        for (index_cnt = 0; index_cnt < 16; index_cnt++) {
                printf("%2x ", output3[index_cnt]);
        }
} else {
        printf("write key success [%d] set\r\n", (int)(key_set + 1));
}
```

.1.5.2.4. Get secure enable or not ? Key set field N is programmed or not API
(透过 API 得知 Secure mode 是否激活, 每組 Key 欄位是否已經寫過 Key)

[Description]

Obtain secure boot enable or not? Specific key field N was programmed or not

[Function]

Function	Description
UINT32 efuse_is_secure_en(void)	Secure boot enable or not
INT32 efuse_key_get_flag(EFUSE_OTP_KEY_SET_FIELD key_set_index)	Specific key set N was programmed or not

[Parameter]

Parameter	Description
EFUSE_OTP_KEY_SET_FIELD	Get specific key field to check if programmed or not

[Return value]

Value	Description
1	Enabled / Programmed

0	Not enabled / not programmed
Others	Error occurred

[See also]

crypto.h

- **Example**

  Reference uboot command

```
// Check if secure enable
int do_nvt_secure_en_cmd(cmd_tbl_t *cmdtp, int flag, int argc, char *const argv[])
{
	if (strncmp(argv[1], "enable", 6) == 0){
		efuse_secure_en();
	} else if(strncmp(argv[1], "get", 3) == 0) {
		if(efuse_is_secure_en())
			printf("secure mode	-> enabled\r\n");
		else
			printf("secure mode	-> disable\r\n");
	} else {
		return CMD_RET_USAGE;
	}
	return 0;
}
U_BOOT_CMD(
	nvt_secure_en,	6,	0,	do_nvt_secure_en_cmd,
	"secure enable (get if enable)\n",
	"[enable] for enable secure mode\n[get] for get if secure mode enable\nAfter secure enable, plantext loader MUST change to cypher loader\n"
);
```

```
// Check specific key field was programmed or not

int do_nvt_key_get_flag_en_cmd(cmd_tbl_t *cmdtp, int flag, int argc, char *const argv[])

{

	UINT32 _key;

	UINT32		key_set = EFUSE_OTP_2ND_KEY_SET_FIELD;

	if (strncmp(argv[1], "0", 1) == 0) {

		key_set = EFUSE_OTP_1ST_KEY_SET_FIELD;

	} else if (strncmp(argv[1], "1", 1) == 0) {

		key_set = EFUSE_OTP_2ND_KEY_SET_FIELD;

	} else if (strncmp(argv[1], "2", 1) == 0) {

		key_set = EFUSE_OTP_3RD_KEY_SET_FIELD;

	} else if (strncmp(argv[1], "3", 1) == 0) {

		key_set = EFUSE_OTP_4TH_KEY_SET_FIELD;

	} else if (strncmp(argv[1], "all", 3) == 0) {

		for(_key = EFUSE_OTP_1ST_KEY_SET_FIELD; _key < EFUSE_OTP_TOTAL_KEY_SET_FIELD; _key++) {

			if(efuse_key_get_flag(_key) == 1)

			{

				printf("key [%d] set flag [O]\r\n", (int)(_key));

			}

			else

			{

				printf("key [%d] set flag [X]\r\n", (int)(_key));

			}

		}

		return 0;

	} else {

		return CMD_RET_USAGE;

	}


	if(efuse_key_get_flag(key_set) == 1)

	{

		printf("key [%d] set flag [O]\r\n", (int)(key_set));

	}
```

```
        else
        {
                printf("key [%d] set flag [X]\r\n", (int)(key_set));
        }
        return 0;
}



U_BOOT_CMD(
        nvt_key_get_flag_en,   6,   0,   do_nvt_key_get_flag_en_cmd,
        "key get flag enable",
        "Represent specific key set already, [all] for all set\n"
);
```

1.5.3. Linux kernel – encrypt and decrypt via crypto framework (with OTP key manager)

The Linux kernel offers a rich set of cryptographic ciphers. The Crypto API is a cryptography framework in the Linux kernel. The NT96680 secure engine drivers develop based on this framework. The Figure 1.3.2-1 is the OpenSSL crypto scenario example from user space to kernel base hardware/software crypto engine driver. The Figure 1.3.2-2 is the kernel crypto framework architecture.

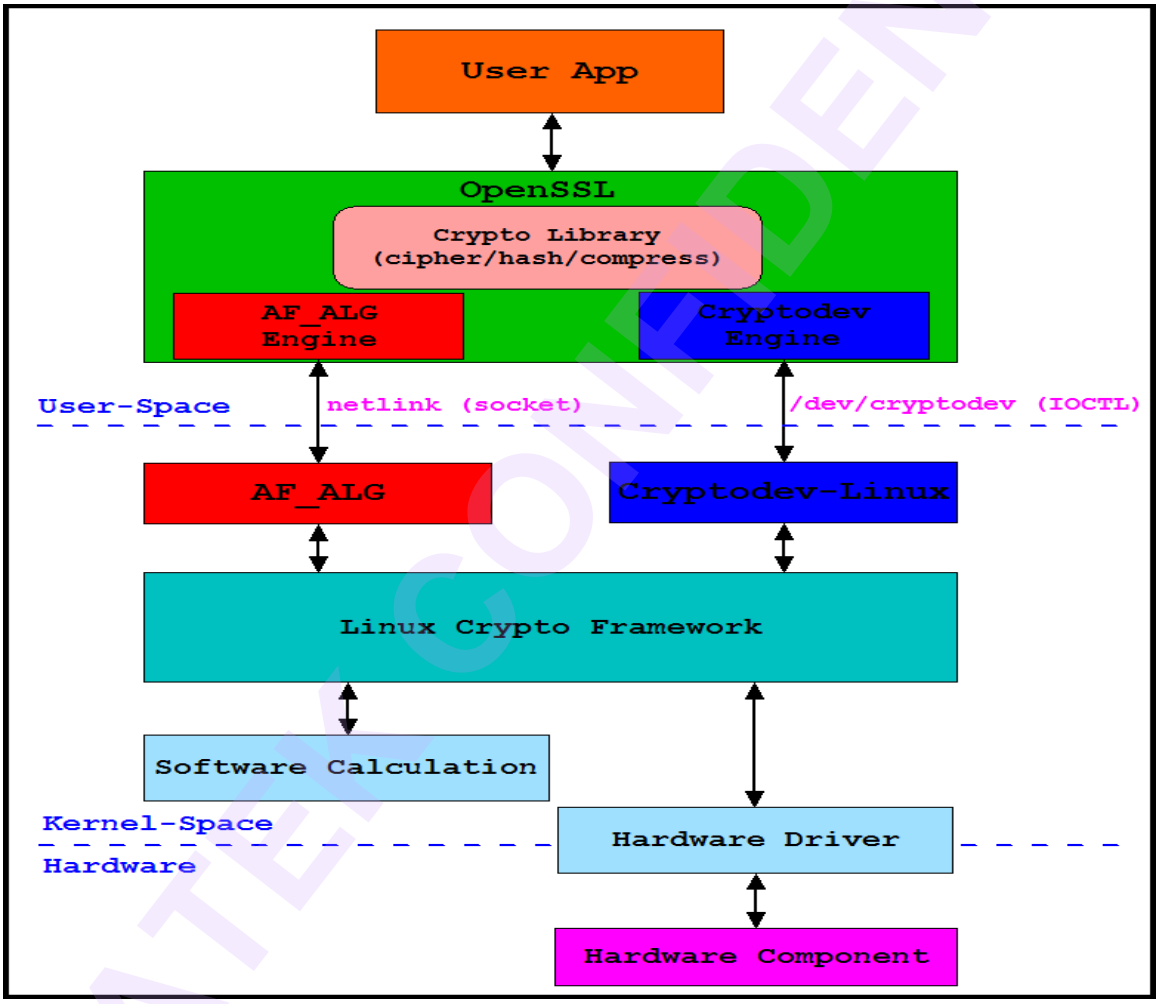


Figure 1.3.2-1 crypto scenario

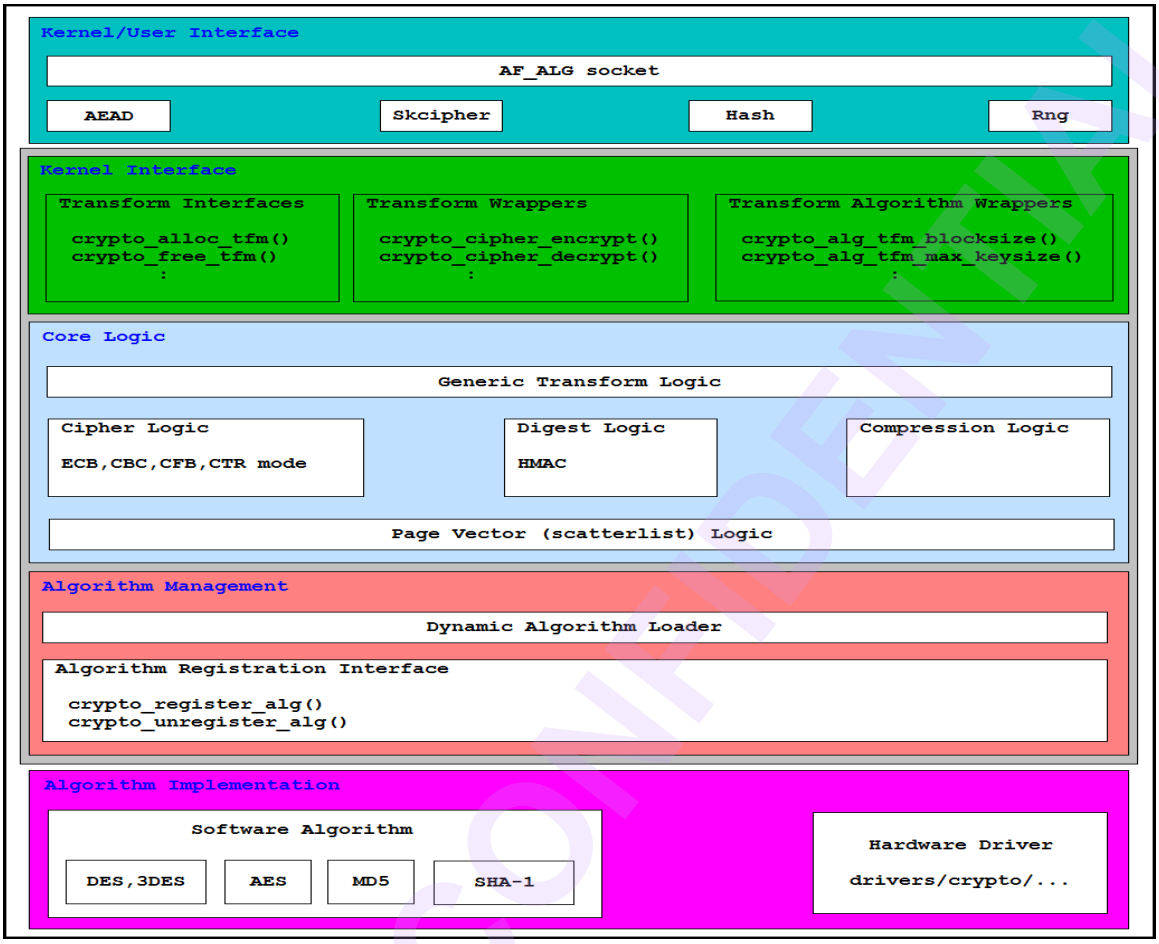
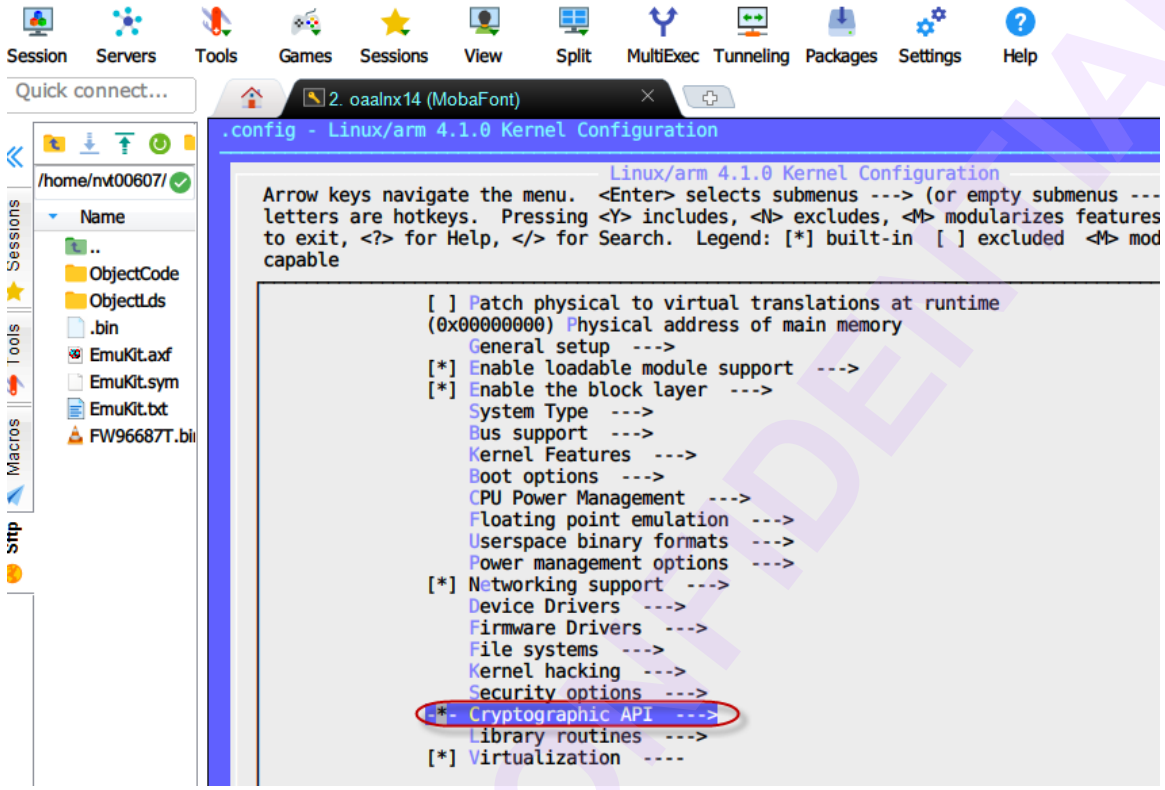
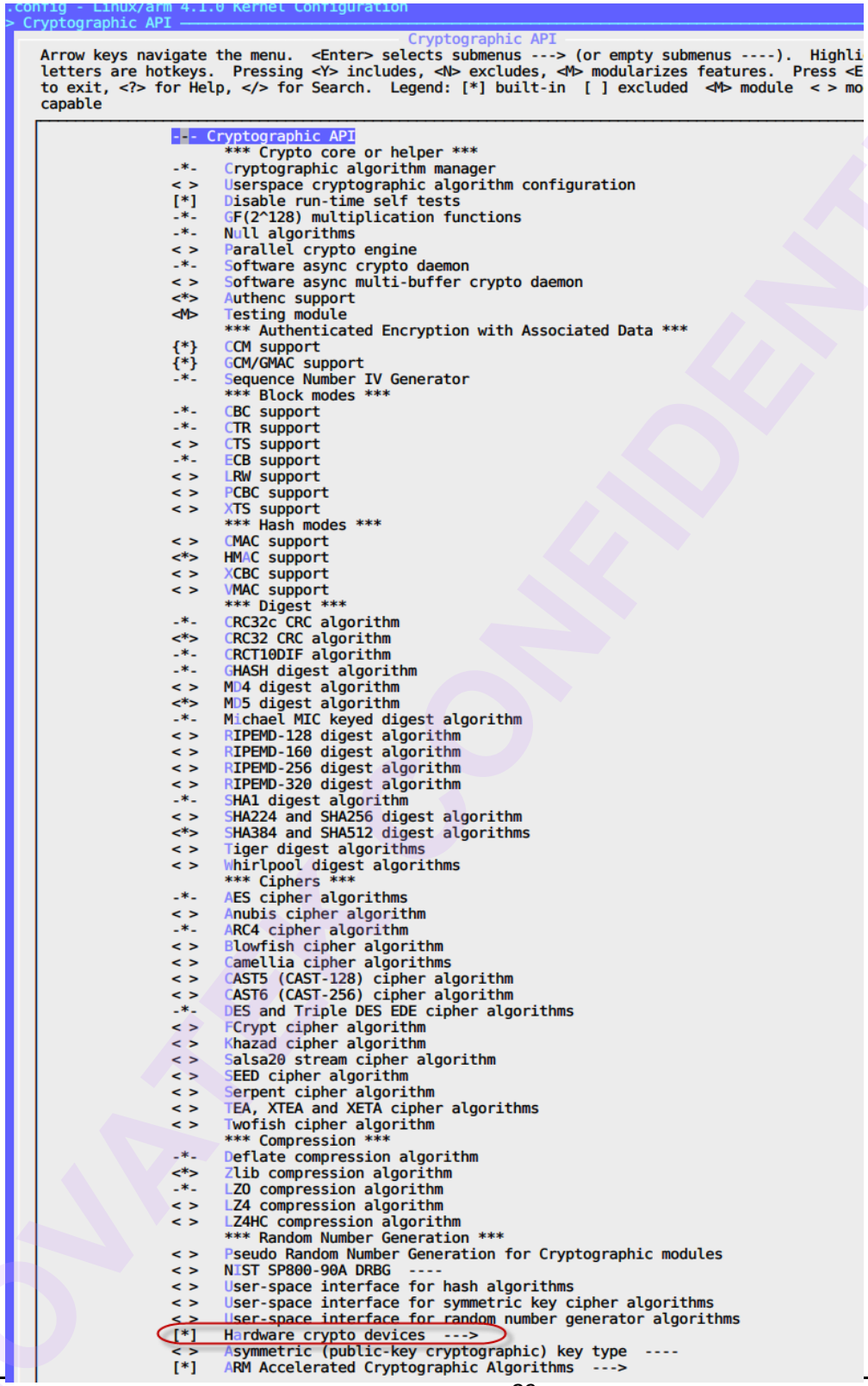


Figure 4-2 Crypto Framework Architecture

.1.5.3.1. Make sure crypto HW engine enabled in linux kernel





.1.5.3.2. Encrypt & Decrypt operation by OTP key

Below is the example from kernel document for cipher operation via key manager in kernel space.

Write an proc sample here

na51000_linux_sdk_181126_commit\na51000_linux_sdk\BSP\linux-kernel\drivers\pinctrl\novatek\nvt_pinmux_proc.c

```
static ssize_t nvt_pinmux_proc_efuse_otp_write(struct file *file, const char __user *buf, size_t size, loff_t *off)
{
	int len = size;
	char cmd_line[MAX_CMD_LENGTH];
	char *cmdstr = cmd_line;
	const char delimiters[] = {' ', 0x0A, 0x0D, '\0'};
	char *argv[MAX_ARG_NUM] = {0};
	unsigned char ucargc = 0;
	UINT32		key_set = EFUSE_OTP_2ND_KEY_SET_FIELD;
	char *scratchpad = NULL;
	int result;
	UINT32	usage;
	struct ablkcipher_def ablk;
	struct ablkcipher_request *req = NULL;
	struct crypto_ablkcipher *tfm = NULL;
	/*check command length*/
	if ((!len) || (len > (MAX_CMD_LENGTH - 1))) {
		pr_err("Command length is too long or 0!\n");
		goto ERR_OUT;
	}
	/*copy command string from user space*/
	if (copy_from_user(cmd_line, buf, len)) {
		goto ERR_OUT;
	}
	cmd_line[len - 1] = '\0';
	printk("CMD:%s\n", cmd_line);
	/*parse command string*/
	for (ucargc = 0; ucargc < MAX_ARG_NUM; ucargc++) {
		argv[ucargc] = strsep(&cmdstr, delimiters);
```

```
		if (argv[ucargc] == NULL) {
			break;
		}
	}
	if (ucargc < 1) {
		pr_err("NULL command error\n");
		goto ERR_OUT;
	}
	if (!strcmp(argv[0], "keyset")) {
		tfm = crypto_alloc_ablkcipher("ecb(aes)-efuse", 0, 0);
		if (!strcmp(argv[1], "0")) {
			pr_info("key set 0\r\n");
			key_set = EFUSE_OTP_1ST_KEY_SET_FIELD;
		} else if (!strcmp(argv[1], "1")) {
			pr_info("key set 1\r\n");
			key_set = EFUSE_OTP_2ND_KEY_SET_FIELD;
		} else if (!strcmp(argv[1], "2")) {
			pr_info("key set 2\r\n");
			key_set = EFUSE_OTP_3RD_KEY_SET_FIELD;
		} else if (!strcmp(argv[1], "3")) {
			pr_info("key set 3\r\n");
			key_set = EFUSE_OTP_4TH_KEY_SET_FIELD;
		}
		if (IS_ERR(tfm)) {
			result = PTR_ERR(tfm);
			printk(KERN_ERR "E: can't load otp(keymanager): % d\n", (int)result);
			goto ERR_OUT;
		} else {
			printk("otp(keymanager)=> cra_name:% s cra_driver_name:%s\n",
				crypto_tfm_alg_name(crypto_ablkcipher_tfm(tfm)),
				crypto_tfm_alg_driver_name(crypto_ablkcipher_tfm(tfm)));
```

```
req = ablkcipher_request_alloc(tfm, GFP_KERNEL);


if (IS_ERR(req)) {
        pr_info("could not allocate request queue\n");
        result = PTR_ERR((void *)req);
        goto ERR_OUT;
} else {
        printk("ablkcipher_request_alloc success\n");
}



ablkcipher_request_set_callback(req, CRYPTO_TFM_REQ_MAY_BACKLOG,
                                                        nvt_ablkcipher_cb,
                                                        &ablk.result);


printk("ablkcipher_request_set_callback success\n");
crypto_ablkcipher_setkey(tfm, NULL, key_set);
printk("ablkcipher_request_setkey success\n");
/* Input data will be random */
scratchpad = kmalloc(16, GFP_KERNEL);
if (!scratchpad) {
        pr_info("could not allocate scratchpad\n");
        goto ERR_OUT;
}
memcpy((void *)scratchpad, (void *)key_2nd, (UINT32)16);
ablk.tfm = tfm;
ablk.req = req;
printk("scratchpad 0x%08x 0x%08x 0x%08x 0x%08x\n", *(UINT32 *)(scratchpad), *(UINT32 *)(scratchpad + 4), *(UINT32 *)(scratchpad +
                8), *(UINT32 *)(scratchpad + 12));
printk("scratchpad 0x%08x \n", (UINT32)scratchpad);
/* We encrypt one block */
sg_init_one(&ablk.sg, scratchpad, 16);
printk("scratchpad 0x%08x 0x%08x 0x%08x 0x%08x\n", *(UINT32 *)(scratchpad), *(UINT32 *)(scratchpad + 4), *(UINT32 *)(scratchpad +
```

```
                8), *(UINT32 *)(scratchpad + 12));


        printk("page_link 0x%08x 0x%08x 0x%08x 0x%08x\n", *(UINT32 *)(sg_virt(&ablk.sg)), *(UINT32 *)(sg_virt(&ablk.sg) + 4), *(UINT32
            *)(sg_virt(&ablk.sg) + 8), *(UINT32 *)(sg_virt(&ablk.sg) + 12));


        ablkcipher_request_set_crypt(req, &ablk.sg, &ablk.sg, 16, NULL);
        printk("ablkcipher_request_set_crypt\n");
        ablkcipher_request_set_tfm(req, tfm);
        printk("ablkcipher_request_set_tfm\n");


        init_completion(&ablk.result.completion);


        /* encrypt data */
        result = nvt_ablkcipher_encdec(&ablk, 1);


        printk("nvt_ablkcipher_encdec done result = %d\n", result);
        if (result) {
                goto ERR_OUT;
        } else {
                pr_info("Encryption triggered successfully\n");
                pr_info("Encrypt => 0x% 08x 0x%08x  0x%08x 0x%08x \n", *(UINT32 *)(sg_virt(&ablk.sg)), *(UINT32 *)(sg_virt(&ablk.sg) + 4), *(UINT32 *)(sg_virt(&ablk.sg) + 8),
*(UINT32 *)(sg_virt(&ablk.sg) + 12));
        }


        /* decrypt data */
        result = nvt_ablkcipher_encdec(&ablk, 0);


        printk("nvt_ablkcipher_encdec done result = %d\n", result);
        if (result) {
                goto ERR_OUT;
        } else {
                pr_info("Decryption triggered successfully\n");
```

```
                pr_info("Decrypt => 0x % 08x 0x%08x 0x%08x 0x%08x\n", *(UINT32 *)(sg_virt(&ablk.sg)), *(UINT32 *)(sg_virt(&ablk.sg) + 4), *(UINT32
                        *)(sg_virt(&ablk.sg) + 8), *(UINT32 *)(sg_virt(&ablk.sg) + 12));
                }
ERR_OUT:
                if (tfm) {
                        crypto_free_ablkcipher(tfm);
                }
                if (req) {
                        ablkcipher_request_free(req);
                }
                if (scratchpad) {
                        kfree(scratchpad);
                }
        }
        pr_err("crypto_alloc_cipher set key proc done\r\n");
        return size;
    }

    return size;
}
```

.1.5.3.3. Get key field is programmed key already or not (透过 API 得知每组 Key 字段是不是已经写过 Key)

[Description]

Obtain specific key field N was programmed or not

[Function]

Function	Description
INT32 efuse_key_get_flag(EFUSE_OTP_KEY_SET_FIELD key_set_index)	Specific key set N was programmed or not

[Parameter]

Parameter	Description

EFUSE_OTP_KEY_SET_FIELD	Get specific key field to check if programmed or not

[Return value]

Value	Description
1	Enabled / Programmed
0	Not enabled / not programmed
Others	Error occurred

Write an proc sample here

na51000_linux_sdk_181126_commit\na51000_linux_sdk\BSP\linux-kernel\drivers\pinctrl\novatek\nvt_pinmux_proc.c

cat this proc for help

```
        return size;
    } else if (!strcmp(argv[0], "getkeyflag")) {
        UINT32  _key;
        UINT32  key_set = EFUSE_OTP_2ND_KEY_SET_FIELD;
        if (!strcmp(argv[1], "0")) {
            key_set = EFUSE_OTP_1ST_KEY_SET_FIELD;
        } else if(!strcmp(argv[1], "1")) {
            key_set = EFUSE_OTP_2ND_KEY_SET_FIELD;
        } else if(!strcmp(argv[1], "2")) {
            key_set = EFUSE_OTP_3RD_KEY_SET_FIELD;
        } else if(!strcmp(argv[1], "3")) {
            key_set = EFUSE_OTP_4TH_KEY_SET_FIELD;
        } else if(!strcmp(argv[1], "all")) {
            for (_key = EFUSE_OTP_1ST_KEY_SET_FIELD; _key < EFUSE_OTP_TOTAL_KEY_SET_FIELD; _key++) {
                if (efuse_key_get_flag(_key) == 1) {
                    pr_info("key [%d] set flag [O]\r\n", (int)(_key));
                } else {
                    pr_info("key [%d] set flag [X]\r\n", (int)(_key));
                }
            }
            return size;
        }

        if (efuse_key_get_flag(key_set) == 1) {
            pr_info("key [%d] set flag [O]\r\n", (int)(key_set));
        } else {
            pr_info("key [%d] set flag [X]\r\n", (int)(key_set));
        }


    } else if (!strcmp(argv[0], "keyset")) {

    }

    return size;
}

static int nvt_pinmux_proc_efuse_help_show(struct seq_file *sfile, void *v)
{
    seq_printf(sfile, "\nUsage\n");
    seq_printf(sfile, "\necho [command] > efuse_op\n\n");
    seq_printf(sfile, "[command]    =>\n");
    seq_printf(sfile, "             => trim (driver's trim data)\n");
    seq_printf(sfile, "             => keyset X (X=0~3)\n");
    seq_printf(sfile, "             => getkeyflag X (X=0~3), all for all field\n");
    seq_printf(sfile, "             => writekeyset X (X=0~3)\n");

    return 0;
}
```