Freescale Semiconductor
Application Note

Document Number:AN4605
Rev. 0, 10/2012

Secure Bootloader Implementation

by: Derek Lau

Contents

1 Introduction

Firmware upgrade is an essential feature of many microcontroller systems. It is very important to encrypt firmware to protect the intellectual property especially when manufacturers provide firmware to third parties.

Freescale has provided broad examples of bootloaders for microcontrollers. This application note describes the implementation of the advanced encryption standard (AES) on the following two USB mass storage bootloaders:

- USB Mass Storage Host Bootloader. See AN4368: USB Mass Storage Host Bootloader, available on **freescale.com**
- USB Mass Storage Device Bootloader. See AN4379: Freescale USB Mass Storage Device Bootloader, available on **freescale.com**

.

2 Implementation of AES

AES is a symmetric key algorithm and the same key is used for encryption and decryption. It is a block cipher which means each time it encrypts or decrypts a block of data. The block size is 128 bits with an optional key size of 128, 192 or 256 bits. The block size of 128 bits is used in the AES implementation given in this application note. Random initial

vector (IV) with the same size as the block is used to further randomize the input data. Without IV, a given block always gets encrypted the same with a given key. The first input block of data is randomized by the IV and then after encryption, it becomes the output block. The other input blocks are randomized by their previous output blocks. The method of randomization depends on the mode selected. For example, in CBC mode, an input block is XOR with the previous output block.

The following figure shows the block diagram of AES encryption and decryption.

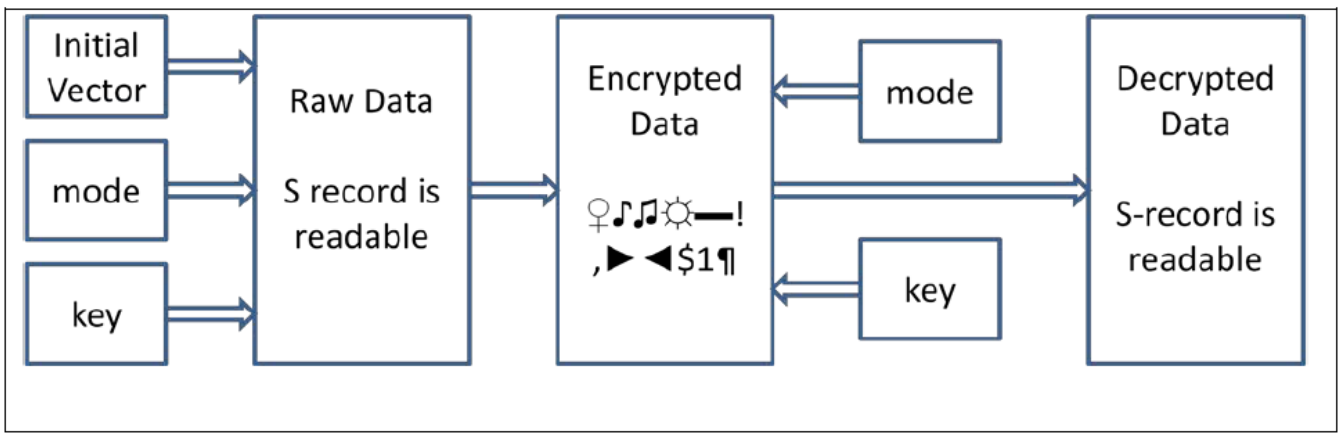

Figure 1. AES encryption and decryption

In this implementation, a PC software is provided to encrypt s-record or binary files. The encrypted files are read and decrypted by the bootloader firmware.

2.1 PC software

There are two files in the PC_Software directory

- encryptfile.exe: The PC software, encryptfile.exe, is used to encrypt s-record and binary files, and generate IV and keys. When the users click the Gen Key File button and type in a file name, an ASCII file containing 128 bits of key will be generated. See Figure 2
- test.key: The test.key file containing the key of f40fd1791254b7f22bd8cdd105aa1d7e means the first 8-bit of key is 0xf4, the second is 0x0f and the last is 0x7e. The images files provided are encrypted by the test key.

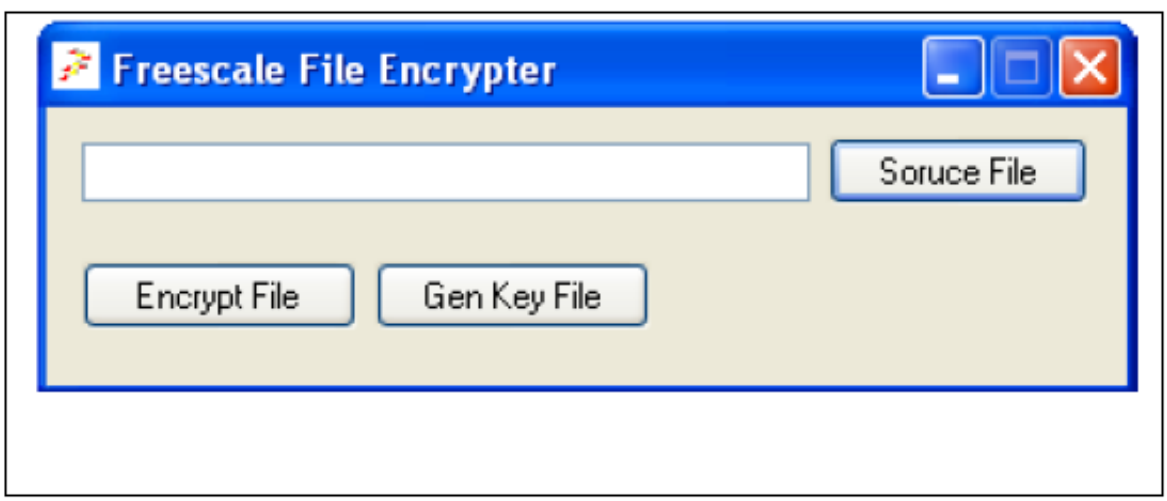

Figure 2. Encryptfile.exe

The first step to encrypt a file is to select the source s-record or binary file by clicking the Source File button. The users can click the Encrypt File button and then choose a key file and an output file to encrypt a file. The key file does not necessarily be generated by the PC software. The user can type in an ASCII key file with the the same format as the test key.

3 AES bootloader firmware

AES decryption is added to the two USB mass storage bootloaders titled:

- Freescale USB Mass Storage Host Bootloader (See AN4368, available on **freescale.com**)
- Freescale USB Mass Storage Device Bootloader (See AN4379, available on **freescale.com**).

The AES decryption uses the crypto acceleration unit library CAU and MMCAU named CAU_MMCAU_SW which can be downloaded from **freescale.com**. The users can add AES or other decryption algorithms to their bootloaders according to their requirements. The CAU library is for ColdFire devices while the MMCAU is for Kinetis devices. Make sure the device contains the hardware crypto acceleration unit when using the crypto library.

3.1 Add AES to bootloader

Follow the procedures below to add AES to a bootloader:

1. Add the folder decrypt and its files to the bootloader project.
2. For ColdFire device, add the cau_lib.a file to the project.

   For Kinetis device, add the mmcau_lib.a file to the project.

   For Codewarrior project, choose File > Properties > C/C++ General > Paths and Symbols > Libraries, to add the library file.

   For IAR project, add the library file to the project in the same way as a .c or .h file.
3. Add the decrypt folder path to the include directory of the project.

   For Codewarrior project, choose File > Properties > C/C++ General > Paths and Symbols > Includes, to add the path.

   For IAR project, choose Project > Options > C/C++ Compiler > Preprocessor to add the path.
4. For Kinetis device only, define FREESCALE_MMCAU.

   For Codewarrior project, choose File > Properties > Symbols, to define the symbol.

   For IAR project, choose Project > Options > C/C++ Compiler > Preprocessor > Defined symbols, to define the symbol.
5. Add the following code line in the file containing the function main:

   ```
   #include aes.h
   ```
6. Call the function aes_main() before the function Flash_Application().

   For example of the Kinetis USB MSD Host Bootloader:

   In the load_image() function in the main.c file, the code becomes:

   ```
   result = aes_main(buffer,&size);
   if (result==0)
   {
       result = FlashApplication(buffer,size);  /* parse and flash an array to flash
   memory */
   }
   ```

   For example of the Kinetis USB MSD Device Bootloader: In the MSD_Event_Callback() function in the disk.c file, the code becomes:

```
error = aes_main(lba_data_ptr->buff_ptr,&(lba_data_ptr->size));
if (!error)
{
        error = FlashApplication(lba_data_ptr->buff_ptr,lba_data_ptr->size);
}
```

7. Use encryptfile.exe to generate a key file.
8. Use encryptfile.exe to encrypt s-record or binary files.
9. Change the key128 in the aes_main() function that generated at step 7. The user must use his/her own key.
10. Compile and run the bootloader according to the instruction given in the application notes AN4368 or AN4379, available on **freescale.com** and use the encrypted s-record or binary file that generated at step 8. Please note that the examples provided by the USB MSD Host Bootloader are using the OSBDM virtual serial port. The old driver uses USB COM port but the new driver will assign a dedicated COM port to the virtual serial port which can be checked from the Windows control panel.
11. This application note comes with a software package that had added AES decryption to the two USB mass storage bootloaders.

4 Customization

The following factors must be considered when implementing crypto bootloader:

- Algorithm to use (For the examples given in this application note, use AES)
- Key length (For the examples given in this application note, use 128-bit)
- Block size (For the examples given in this application note, use 128-bit
- Cipher mode (For the examples given in this application note, use CBC)

The encryption and decryption must use the same method and parameters. When sending encrypted data to the microcontrollers, it is more convenient to send data with length equal to the multiples of the block size. The decryption must be done in the firmware but not in the PC software, otherwise data can be captured from the physical bus such as USB.

5 Conclusion

AES is implemented on two USB mass storage bootloaders to show AES decryption in bootloader firmware. A PC software is provided to generate key file and IV vectors, and encrypt s-record and binary files. The users can easily add encryption in their bootloaders using AES or any other crypto algorithms to fit their requirements.

6 References

The following reference documents are available at **freescale.com**.

- AN4368: USB Mass Storage Device Host Bootloader
- AN4379: USB Mass Storage Device Bootloader
- CAUAPIUG: CAU and mmCAU API User Guide

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: AN4605
Rev. 0, 10/2012