AN12527

LPC55Sxx PRINCE Real-time Data Encryption

Rev. 1 — 26 May 2020 Application Note

1 PRINCE introduction

The PRINCE algorithm is used for real-time encrypt/decrypt operation on LPC55Sxx on-chip flash contents. PRINCE is fast compared to AES because it can decrypt and encrypt without adding extra latency. PRINCE operates as data is read or written to flash without the need to first store data in RAM and then encrypt or decrypt to another memory space. PRINCE operates on block of 64 bits with a 128bit key size.

This functionality is useful for asset protection, such as securing application code, securing data, and enabling secure flash update.

The on-chip flash is divided into three regions for encryption/decryption. These regions are referred to as crypto regions. The LPC55S6x/LPC55S2x/LPC552x supports three regions for encryption and decryption, referred to as crypto regions. See Figure 185 below. Each crypto region resides at a 256 kB address boundary within the flash. For the 640 kB flash size in LPC55S6x/ LPC55S2x/LPC552x, the first two crypto regions are 256 kB in size, and the third one is 128 kB. For the other flash size, the region range can be set by configure the register BASE_ADDRn bits[19:18]. If the bit[19:18] is 0x0 in BASE_ADDR1, the region1 will cover from 0x0 to 0x3FFFF address flash. In this application note, 640 kB flash size as an example, every region covers different address flash range.

Each crypto region is subdivided into 8 kB subregions. PRINCE encryption/decryption can be enabled or disabled for each sub region. The enabled subregions need not be contiguous.

Each crypto region has a dedicated Key and an Initialization Vector (IV). This allows multiple images to reside in the flash with an independent encryption base. The Key is sourced from on-chip SRAM PUF via an internal hardware interface, without exposing the key on the system bus.

Figure 1 shows an example where different PRINCE regions are assigned with different memory areas. The subregions marked with “c” are “crypto” enabled, that is, they are enabled for both encryption and decryption. The gray subregions stand for not used.

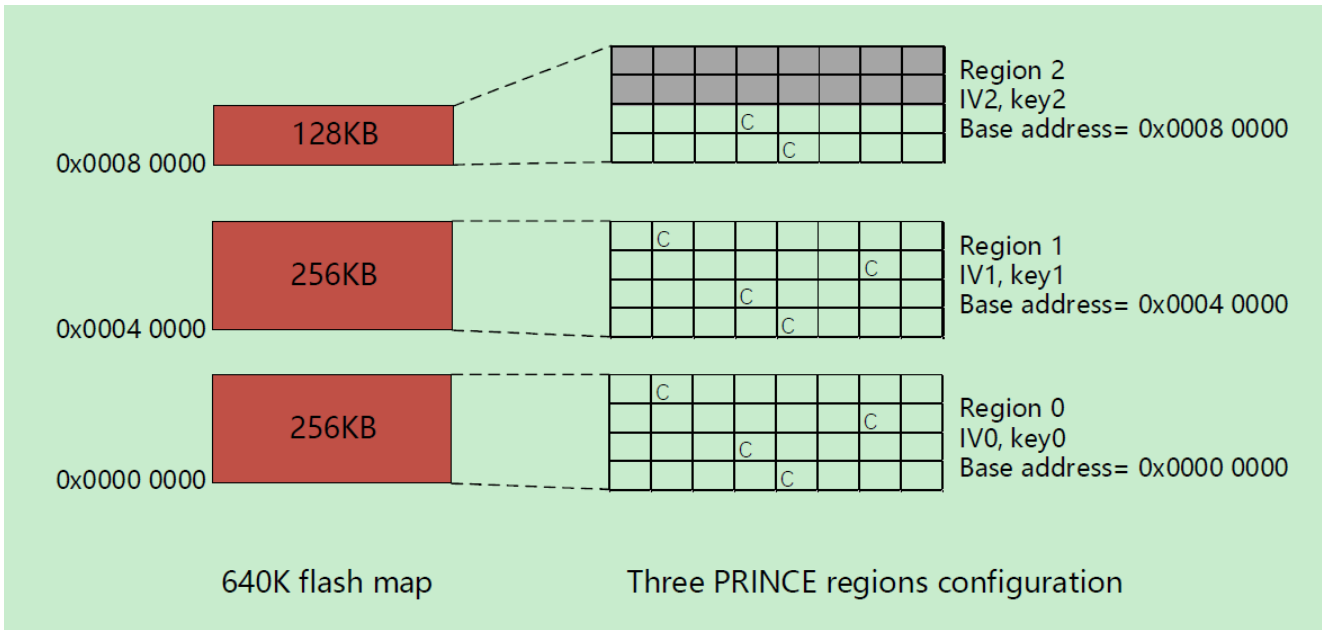


Figure 1. PRINCE configuration to 512 KB memory map

Figure 2 shows an example where region 0 and region 1 cover the same 256 KB memory area. In this way, customer can to use different key to secure the 2nd bootloader and application code in the 256 KB flash size chip.

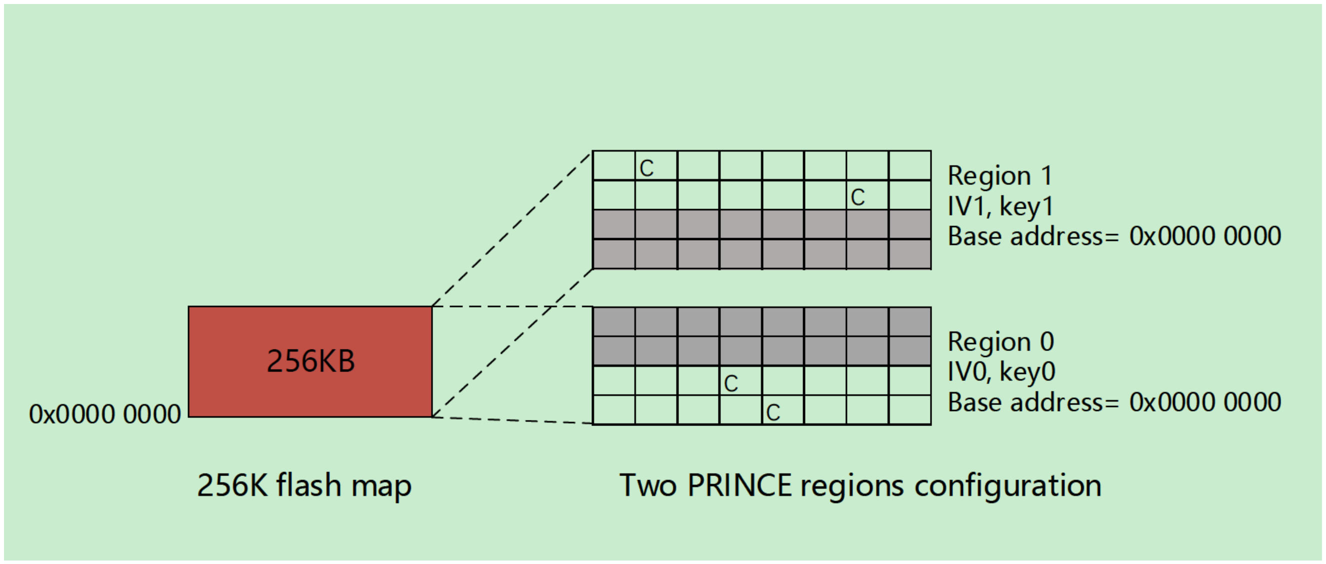

256KB
0x0000 0000
C
Region 1
IV1, key1
Base address= 0x0000 0000
Region 0
IV0, key0
Base address= 0x0000 0000
256K flash map
Two PRINCE regions configuration

Figure 2. PRINCE configuration to 256 KB memory map

2 PRINCE demonstration step-by-step

The keys used for PRINCE encryption/decryption are derived from on-chip SRAM PUF. KeyStore is stored in FFR region of flash at address 0x9E600 that contains the activation code of the device and the key code for PRINCE key of various PRINCE regions. The PRINCE keys are delivered through an internal hardware interface and are not software accessible. On every reset, the boot ROM reads the KeyStore and reconstructs the PRINCE keys into the PRINCE engine.

The BLHOST utility can be used to provision the keys into the LPC55S69 device. During provisioning process the activation code and key code are initially stored in internal SRAM of the device which is later stored onto PFR region.

Enter the device into UART ISP mode and open BLHOST.

2.1 PRINCE-related PUF key store setup

In the following example, you can see the sequence of commands to be issued from PC blhost application to the device in ISP mode to generate proper PRINCE-enabled key store. The key store is saved into device PFR and accessed by boot ROM during secure boot.

WARNING

Perform *key-provisioning enroll* operation only once for each device in the whole lifetime of chip. Ideally, *set_key/ write_key_nonvolatile* operations need better to be performed once in the whole lifetime of chip. In other words, there need not implement these commands again after key provisioning done. PRINCE configuration and flash erasing/programing can be done repeatedly later.

NOTE

The experiments in this application note use the 1B revision silicon.

WARNING

After performing *key-provisioning write_key_nonvolatile* step, the chip must be reset by the reset pin or POR so that the new key can be sent to PRINCE engine successfully.

1. Open the blhost PC tool, connect to the processor using UART (in this example UART is COM108). Make the processor into ISP mode by pressing ISP pin during reset stage.
2. Get the version of bootROM and check the availability of communication.

```
blhost.exe -p COM108 -- get-property 1
```

3. Generate device activation code and store it into key store structure.

```
blhost.exe -p COM108 -- key-provisioning enroll
```

4. Generate random PRINCE region 0. (PRINCE region 0 key type = 7)

```
blhost.exe -p COM108 -- key-provisioning set_key 7 16
```

5. Generate random PRINCE region 1. (PRINCE region 1 key type = 8)

```
blhost.exe -p COM108 -- key-provisioning set_key 8 16
```

6. Generate random PRINCE region 2. (PRINCE region 2 key type = 9)

```
blhost.exe -p COM108 -- key-provisioning set_key 9 16
```

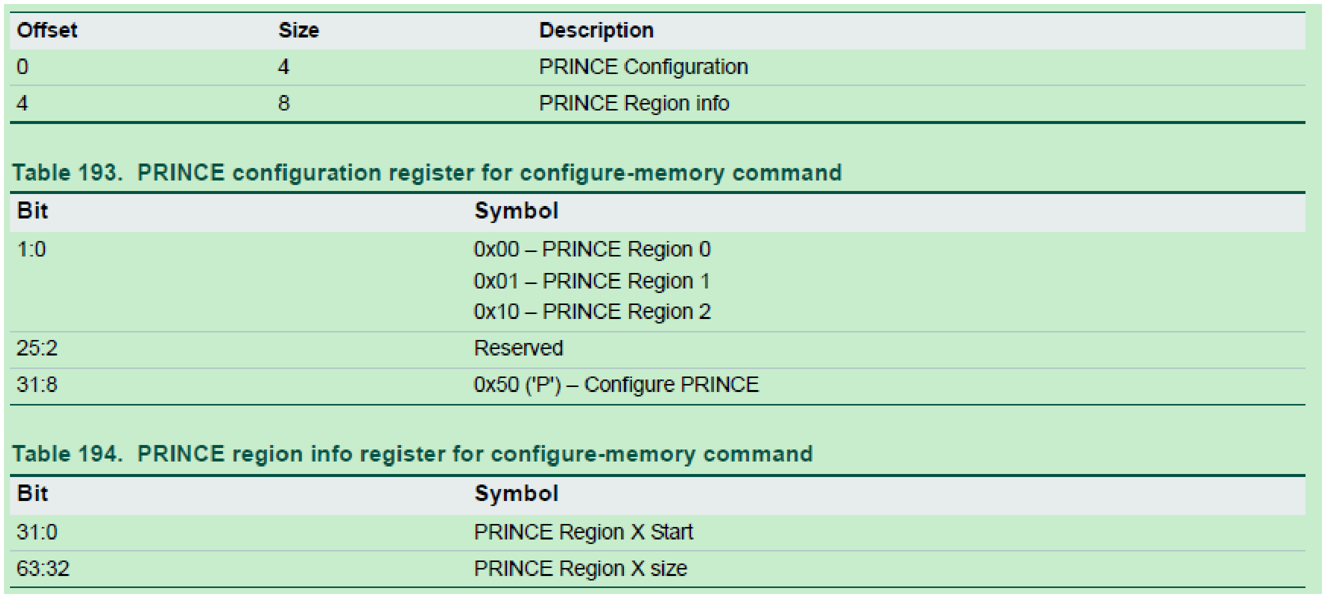
7. Save the key store into PFR page of Flash memory

```
blhost.exe -p COM108 -- key-provisioning write_key_nonvolatile 0
```

8. Press reset pin or POR to reset the device.

2.2 PRINCE region configuration

For PRINCE encryption and decryption, the regions and subregions for the crypto operation are configured. This can be done with ISP command “configure-memory”. This command must be called with following data structure.



Offset	Size	Description
0	4	PRINCE Configuration
4	8	PRINCE Region info

Table 193. PRINCE configuration register for configure-memory command

Bit	Symbol
1:0	0x00 – PRINCE Region 0 0x01 – PRINCE Region 1 0x10 – PRINCE Region 2
25:2	Reserved
31:8	0x50 ('P') – Configure PRINCE

Table 194. PRINCE region info register for configure-memory command

Bit	Symbol
31:0	PRINCE Region X Start
63:32	PRINCE Region X size

Figure 3. Structure for configure-memory command

Load structure into RAM memory and call "configure-memory" command with this sequence:

WARNING

Length of the encrypted area must be equal to range to be erased later and be equal to range to be programmed later. So, certain pattern must be filled at the end of the binary file you created to be aligned to length.

1. Connect to the processor again using UART (in this example UART is COM108). Make the processor into ISP mode by pressing ISP pin during reset stage.
2. Get the version of bootROM and check the availability of communication.

```
blhost.exe -p COM108 -- get-property 1
```

3. Region selection (Region 0 in this example).

```
blhost.exe -p COM108 -- fill-memory 0x20034000 4 0x50000000
```

4. Start address of encrypted area (Address 0x0 in this example).

```
blhost.exe -p COM108 -- fill-memory 0x20034004 4 0
```

5. Length of the encrypted area (0x10000 in this example).

```
blhost.exe -p COM108 -- fill-memory 0x20034008 4 0x10000
```

6. Call configure-memory with prepared structure in RAM.

```
blhost.exe -p COM108 -- configure-memory 0 0x20034000
```

WARNING

After you finish above configuration commands, do not reset the board and continue with commands for erasing the flash and loading the image.

Following this command, PRINCE is configured for flash encryption.

NOTE

The PFR area should be excluded from PRINCE encryption area. That is, the start and size settings in configuration the structure must be set to avoid overlapping with the PFR area.

2.3 Erase flash and upload image

A "prince erase checker" is implemented in the boot ROM that checks whether the whole PRINCE enabled area which consists of one or more subregions is erased at once. Similarly, "prince flash write checker" is implemented in the ROM code to check whether the whole enabled area which consists of one or more sub-regions is programmed at once. To load the image that is on-the-fly encrypted by PRINCE, the following sequence of ISP commands is issued using blhost tool:

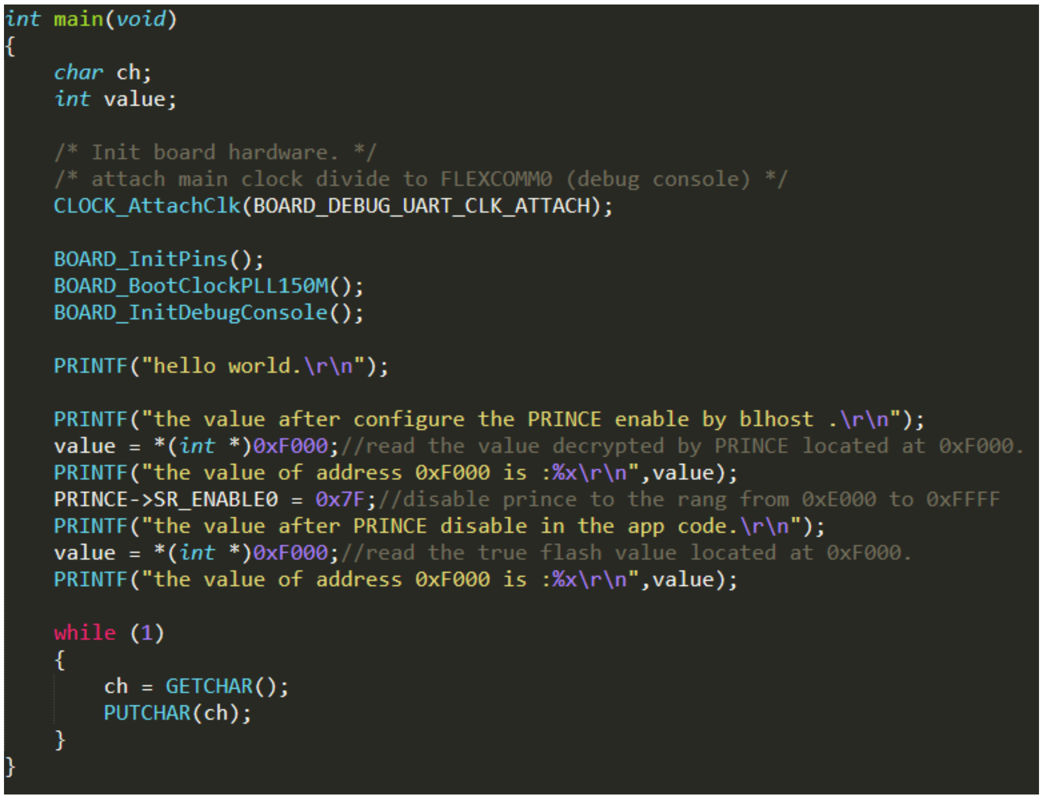
WARNING

If length of the encrypted area is 0x10000 as above, the erase and program region should be set to 0x10000. The binary file size must be 0x10000.

[Preparation]:

Open and compile a LPC55Sxx project, create the binary file. Fill the pattern, that is 0x55, into the binary to 0x10000 bytes size. This example uses a named hello_world_0x10000_size.bin file which comes from SDK and has been enlarged to 0x10000 bytes.

Disable a PRINCE subregion and read the flash value in this subregion. The true flash value is received. This means, PRINCE function can be verified. For details, see Figure 4.



```
int main(void)
{
    char ch;
    int value;

    /* Init board hardware. */
    /* attach main clock divide to FLEXCOMM0 (debug console) */
    CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);

    BOARD_InitPins();
    BOARD_BootClockPLL150M();
    BOARD_InitDebugConsole();

    PRINTF("hello world.\r\n");

    PRINTF("the value after configure the PRINCE enable by blhost .\r\n");
    value = *(int *)0xF000;//read the value decrypted by PRINCE located at 0xF000.
    PRINTF("the value of address 0xF000 is :%x\r\n",value);
    PRINCE->SR_ENABLE0 = 0x7F;//disable prince to the rang from 0xE000 to 0xFFFF
    PRINTF("the value after PRINCE disable in the app code.\r\n");
    value = *(int *)0xF000;//read the true flash value located at 0xF000.
    PRINTF("the value of address 0xF000 is :%x\r\n",value);

    while (1)
    {
        ch = GETCHAR();
        PUTCHAR(ch);
    }
}
```

Figure 4. APP code

1. Erase the flash memory (0x10000 in this example).

```
blhost.exe -p COM108 -- flash-erase-region 0x0 0x10000
```

2. Load the image into the flash.

```
blhost.exe -p COM108 -- write-memory 0 hello_world_0x10000_size.bin
```

3. After these steps, the image loaded in the flash is encrypted.

NOTE

Under the certain condition, successful generic responses may be received when partial erasing and programming with checker commands are sent. This result does not stand for allowing partial erasing and programming and may lead to uncontrollable status. Therefore, the whole prince enabled area must be implemented once.

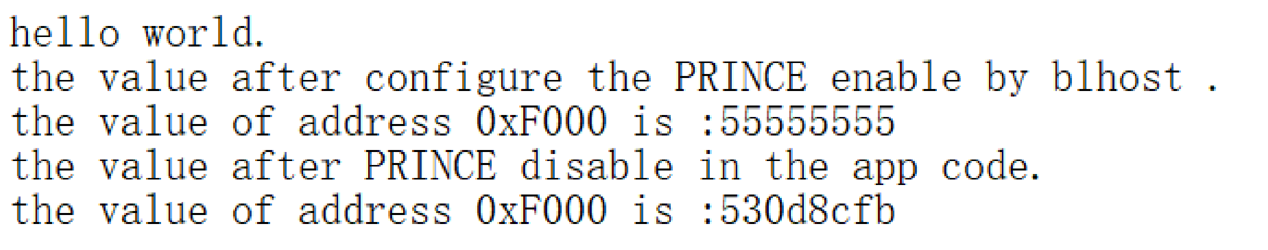
NOTE

The range of erasing and programming must not exceed one region size (256 K bytes). If multiple regions are enabled by PRINCE, erasing and programming is done separately by region by region.

2.4 Run code

1. Connect to the processor again using UART and open the CommAssistant.
2. Press reset pin or POR to reset the device.

   The strings are printed in the Figure 5.



```
hello world.
the value after configure the PRINCE enable by blhost .
the value of address 0xF000 is :55555555
the value after PRINCE disable in the app code.
the value of address 0xF000 is :530d8cfb
```

Figure 5. CommAssistant window

NOTE

The value of address 0xF000 after disabling PRINCE isn't 0x530d8cfb permanently, it is based on certain condition in every experiment.

3 Revision history

This table summarizes changes to this document.

Table 1. Revision history

Rev	Date	Description
0	25 October 2019	Initial version
1	26 May 2020	Section 1 updated

Date of release: 26 May 2020
Document identifier: AN12527