AN12527

LPC55Sxx PRINCE Real-Time Data Encryption

Rev. 4 — 20 September 2023

Application note

Document information

Information	Content
Keywords	AN12527, PRINCE, memory encryption/decryption, security feature, LPC55Sxx, LPC55
Abstract	This application note introduces and demonstrates PRINCE using the LPC55Sxx series of devices.

1 Introduction

The PRINCE algorithm is used to encrypt and decrypt the on-chip flash contents of the LPC55Sxx series of devices in real-time. PRINCE is fast compared to advanced encryption standard (AES) because it can decrypt and encrypt without adding extra latency. PRINCE operates by reading or writing data to flash without storing it in RAM first. PRINCE then encrypts or decrypts this data before moving it to another memory space. PRINCE operates on blocks of 64 bits with a 128-bit key size.

This functionality is useful for asset protection such as securing application code, securing application 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. Each crypto region resides at a 256 kB address boundary within the flash. Each crypto region is subdivided into 8 kB subregions. PRINCE encryption/decryption can be enabled or disabled for each subregion. The subregions, which are enabled do not need to be contiguous.

There are two possible ways for those crypto regions to work which are as follows:

- The three regions can fit into the entire memory. For example, the LPC55S6x, which has 640 kB of total flash and can therefore have a different starting address for each region, see Figure 1.
- The three regions can overlap each other. For example, the LPC55S1x, which has 256 kB of total flash and therefore must have the same starting address for each region, see Figure 2.

For the case where the regions are overlapping, the subregions can be configured per region to ensure that each region has been mapped to physical memory.

Each crypto region has a dedicated key and an initialization vector (IV). As a result, multiple code images can reside in the flash with an independent encryption base. The key is sourced from the 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 can be fitted in the entire physical memory. The subregions marked with “c” are “crypto” enabled, meaning they are enabled for both encryption and decryption. The gray subregions stand for not used.

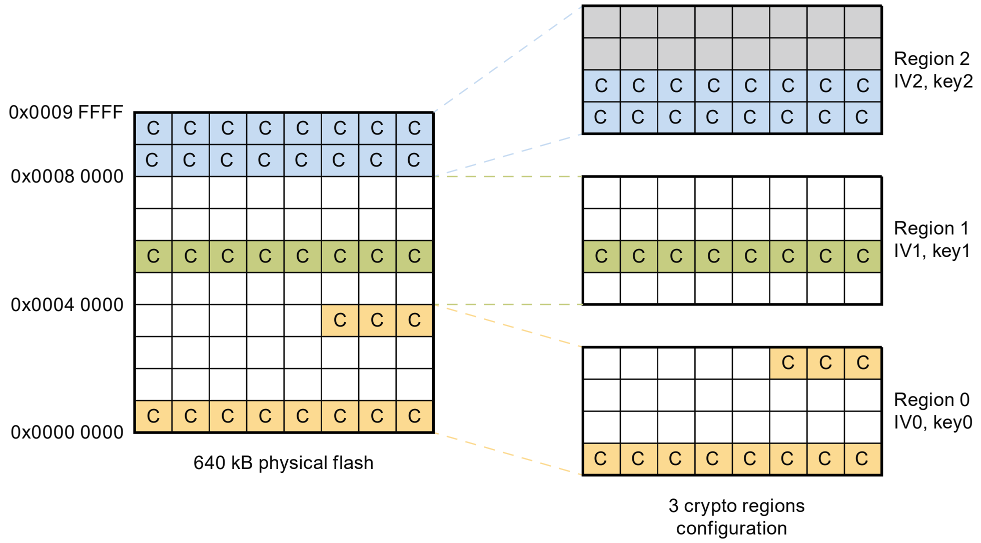


Figure 1. PRINCE configuration for 640 kB physical flash

Figure 2 shows an example where three different crypto regions cover the same 256 kB memory area. This way, the customer can, for example, secure the secondary bootloader and application code in the 256 kB flash-sized chip using a different key.

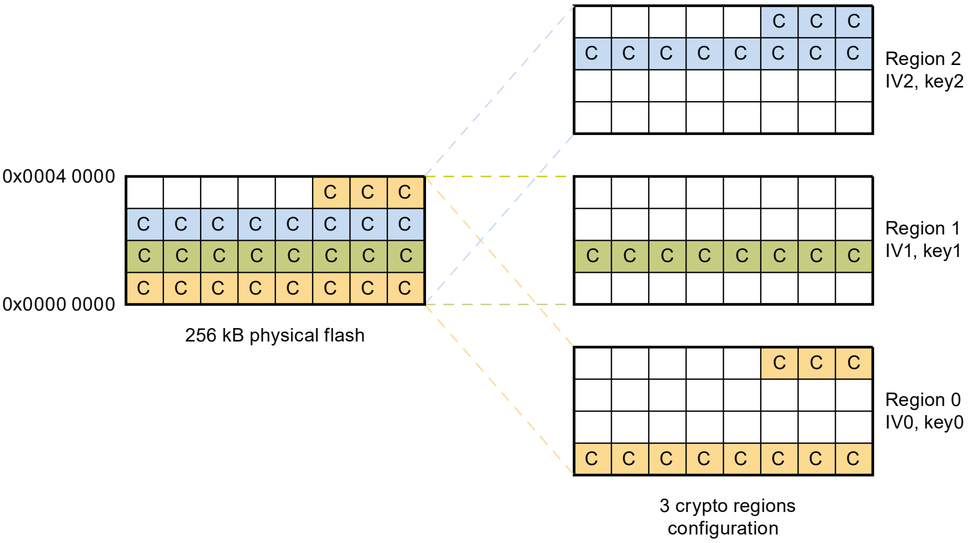


Figure 2. PRINCE configuration for 256 kB physical flash

2 Acronyms

Table 1 lists the acronyms used in this document.

Table 1. Acronyms

Acronym	Meaning
AES	Advanced encryption standard
RAM	Random access memory
IV	Initialization vector
SRAM	Static random access memory
PUF	Physically unclonable function
PFR	Protected flash region
ROM	Read-only memory
ISP	In-system programming
UART	Universal asynchronous receiver-transmitter

3 Step-by-step PRINCE demonstration

The keys used for PRINCE encryption/decryption are derived from on-chip SRAM PUF. The KeyStore resides in the PFR region of flash that contains the activation code of the device and the KeyCode for the 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[1] utility can be used to provision the keys into the LPC55Sxx device. During the provisioning process, the activation code and key code are initially stored in the internal SRAM of the device, which is later stored onto the PFR region.

[1] The blhost utility can be found on the MCUBOOT webpage.

Important: *The following subsections use the 1 B silicon revision of LPC55S69. Memory addresses, configuration, and collateral software differ from other platforms with PRINCE support. For more details, see the product-specific reference documentation/software.*

3.1 PRINCE-related PUF key store setup

To generate a proper PRINCE-enabled key store, the example in this section shows the sequence of commands that must be issued from the PC blhost application to the device in ISP mode. The key store is saved into device PFR and accessed by boot ROM during secure boot.

Warning: *The key provisioning operations (enroll, SetKey, and write_key_nonvolatile) must only be performed once during the lifetime of the chip. PRINCE configuration and flash operations (erase/programming) can be executed multiple times.*

1. To execute blhost commands, open a terminal.
2. Connect to the processor using UART (in this example UART is COM108). Pressing the ISP pin during the reset stage puts the processor into ISP mode.
3. Get the version of boot ROM and check the availability of communication.

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

4. Generate a device activation code and store it into a key store structure.

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

5. Generate a random PRINCE region 0. (PRINCE region 0 uses key type 7 from the PUF)

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

6. Generate random PRINCE region 1. (PRINCE region 1 uses key type 8 from the PUF)

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

7. Generate random PRINCE region 2. (PRINCE region 2 uses key type 9 from the PUF)

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

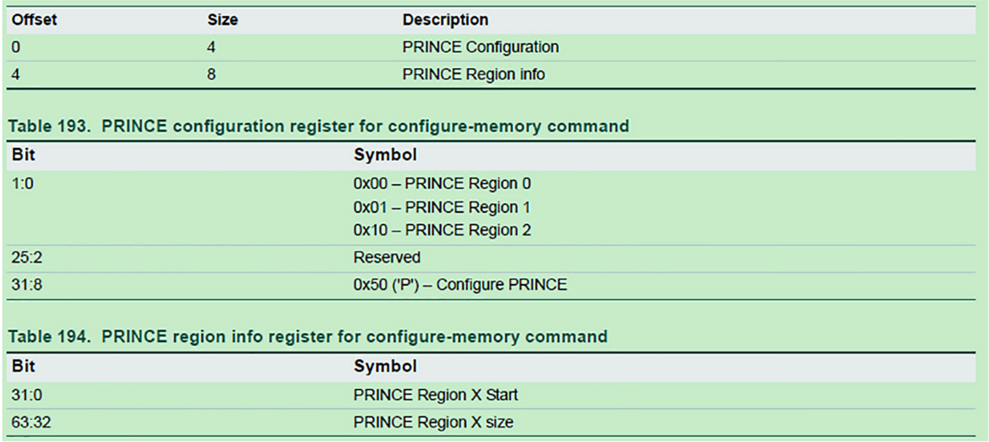
8. Save the key store into the PFR page of flash memory

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

9. To reset the device, press the reset pin or POR.

3.2 PRINCE region configuration

For PRINCE encryption and decryption, the regions and subregions for the crypto operation are configured. This configuration can be done with the ISP command `configure-memory`. This command must be called with the data structure shown in Figure 3.



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 the structure into RAM memory and call the `configure-memory` command with this sequence:

1. Connect to the processor again using UART (in this example UART is COM108). Pressing the ISP pin during the reset stage puts the processor into ISP mode.
2. Get the version of boot ROM 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 the 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 completing the configuration commands listed above, continue with the commands for erasing the flash and loading the image without resetting the board.*

Note: *The PFR area must be excluded from the PRINCE encryption area. Therefore, to avoid overlapping with the PFR area, set the start and size settings in the configuration of the structure.*

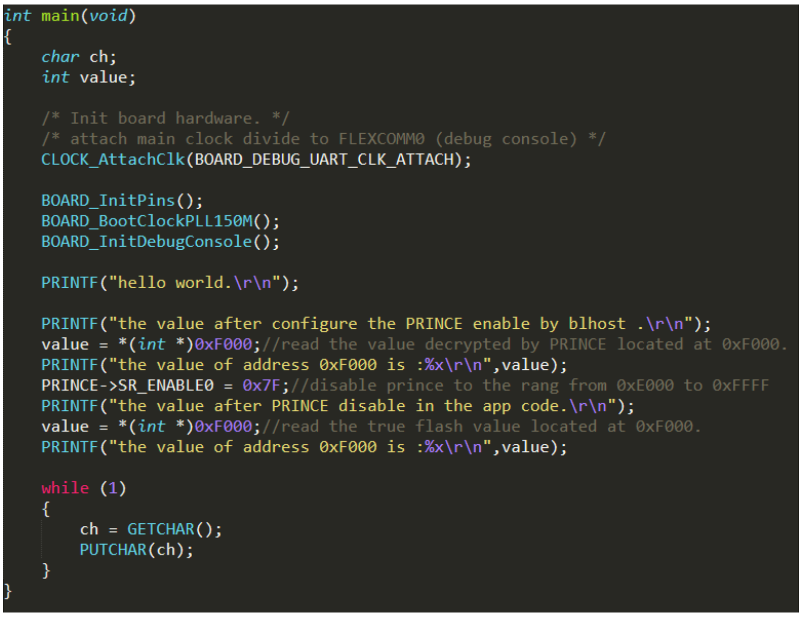
3.3 Erase the flash and upload the image

A "PRINCE erase checker" is implemented in the boot ROM that checks whether the entire PRINCE-enabled area, which consists of one or more subregions, is erased all at once. Similarly, the "PRINCE flash write checker" is implemented in the ROM code to check whether the entire enabled area, which consists of one or more subregions, is programmed all at once. This means that the length used in Step 5 of Section 3.2 must be equal to the size of the binary that is flashed to the board. If the binary is smaller in size, expand it until the desired length is reached. For example, 0x10000 bytes are 64 kB, which is a multiple of 8 kB. This 8 kB is the size of one subregion.

Ensure to adhere to the size set in Step 5 of Section 3.2 and prepare the binary as follows:

- Open and compile a LPC55Sxx project.
- Create the binary file.

- Fill the binary to a size of 0x10000 bytes with the pattern, which is 0x55. This example uses a file from the SDK called `hello_world_0x10000_size.bin` that has been expanded to 0x10000 bytes.
- Disable a PRINCE subregion and read the flash value in this subregion. The true flash value is received, which means that the PRINCE function can be verified. For details, refer Figure 4.



```c
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

To load the image that is on-the-fly encrypted by PRINCE, the following sequence of ISP commands is issued using blhost:

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
   ```

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

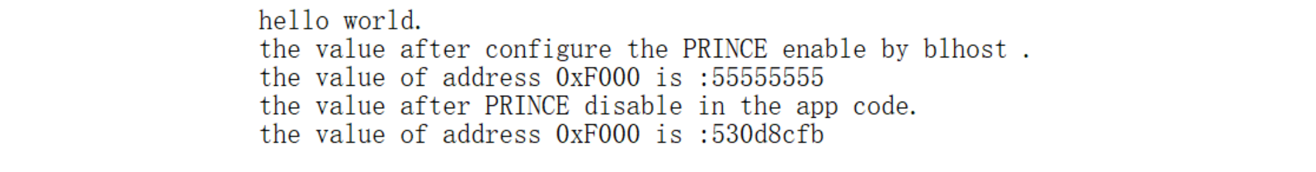
Note: *Under certain conditions, generic success responses can be received when issuing the partial erase and program commands. This response can lead to putting the device in an uncontrollable state. Therefore, it is advised to implement the entire PRINCE-enabled area at once.*

Note: *The range of erasing and programming must not exceed one region size (256 kbit). If PRINCE enables multiple regions, erasing and programming is done separately, region by region.*

3.4 Run code

To run the code, perform the following steps:

1. Connect to the processor again using UART and open a terminal application.
2. To reset the device, press reset pin or POR.
   The strings are printed, as shown in 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 is not always 0x530d8cfb, rather, it depends on specific conditions in each experiment.*

4 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

5 Revision history

Table 2 summarizes the revisions done to this document.

Revision history

Revision history	Release date	Description
4	20 September 2023	• Multiple editorial changes throughout the document • Figures updated to svg format • Spelling and grammar improvements for the entire document • Figure 1 and Figure 2 updated
3	11 May 2021	Added one note in Section 3.3
2	28 October 2020	Replaced LPC55Sxx for LPC55S6x/LPC55S2x/LPC552x
1	26 May 2020	Section 1 updated
0	25 October 2019	Initial public release

6 Legal information

6.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

6.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com

Date of release: 20 September 2023
Document identifier: AN12527