# LPC55S1x

## SESIP Security Target

Rev. 0.9 — 15 September 2020 — Evaluation document

**Document information**

| Information | Content |
|---|---|
| Keywords | SESIP, Security Target, LPC55S1x , LPC55S16, LPC55S14 |
| Abstract | Evaluation of the LPC55S1x developed and provided by NXP Semiconductors, BL Edge Processing, according to SESIP Assurance Level 2 (SESIP2), based on SESIP methodology, version 1.0 |

# Revision History

| Rev. | Date | Description |
|---|---|---|
| 0.1 | 2020-04-29 | First version |
| 0.2 | 2020-05-04 | Updated PP reference version and SDK content |
| 0.3 | 2020-05-05 | Updated PP reference version, Section 2.1, added Section 3.2.5 and explicitly stated RSA mechanism used in Section 3.2.3 and Section 3.2.4. |
| 0.4 | 2020-05-13 | Added ROM Patch Reference |
| 0.5 | 2020-08-19 | Updated according to evaluator feedback |
| 0.6 | 2020-08-28 | Updated according to evaluator feedback |
| 0.7 | 2020-09-10 | Updated debug function and patch version description |
| 0.8 | 2020-09-11 | Updated debug function |
| 0.9 | 2020-09-15 | Updated Section 3.2.1 |

# 1 Introduction

This Security Target describes the LPC55S1x platform and the exact security properties of the platform that are evaluated against GlobalPlatform Technology Security Evaluation Standard for IoT Platforms (SESIP), version 1.0 [1].

## 1.1 ST Reference

LPC55S1x, SESIP Security Target, Revision 0.9, NXP Semiconductors, 15 September 2020.

## 1.2 Protection Profile Reference and Conformance Claims

Table 1. Protection Profile Reference and Conformance Claims

| Reference | Value |
|---|---|
| PP Name | Protection Profile for Secure MCUs and MPUs [2] |
| PP Version | V0.107 |
| Assurance Claim | SESIP Assurance Level 2 (SESIP2) |
| Package Claim | Base PP, Package Secure Services, and Package Software Isolation |

## 1.3 Platform Reference

LPC55S1x

Table 2. Platform Reference

| Reference | Value |
|---|---|
| Platform Name and Version | LPC55S1x, Rev. A0<br>ROM Firmware, 3.0.0<br>ROM Firmware Patch, Rev. 3 |
| Platform Identification | LPC55S14, LPC55S16 |
| Platform Type | Microcontroller platform for IoT applications |

## 1.4 Included Guidance Documents

The following documents are included with the platform:

Table 3. Guidance Documents

| Document | Reference |
|---|---|
| User Manual | UM11295, LPC55S1x/LPC551x User Manual, Rev.1.3, NXP Semiconductors [4] |
| Product Data Sheet | LPC55S1x/LPC551x Product data sheet, Rev.0.7, NXP Semiconductors [5] |
| Application Note | AN12278, LPC55S00 Security Solutions for IoT, Rev. 1, NXP Semiconductors [8] |
| Application Note | AN12283, LPC55Sxx Secure Boot, Rev. 1, NXP Semiconductors [6] |
| Application Note | AN12445, Asymmetric Cryptographic Accelerator CASPER, Rev. 3, NXP Semiconductors [7] |

| Document | Reference |
|---|---|
| Application Note | AN12326, Secure GPIO and Usage, Rev. 1, NXP Semiconductors [9] |
| SESIP Security Target | LPC55S1x, SESIP Security Target, Revision 0.9, NXP Semiconductors, 15 September 2020. |
| API Reference Manual (Optional) | MCUXpresso SDK API Reference Manual Rev. 0 in LPC55S16 SDK 2.8.0 [10] |

## 1.5 Platform Overview and Description

The LPC55S1x consists of hardware and software, the software is divided into an immutable part, stored in ROM, which includes the code performing the MCU/MPU's secure initialization, as well as the optional firmware, which can be modified and updated during the product lifecycle.

The LPC55S1x is intended to be used by an integrator as a basis to develop an IoT Platform, by adding to it the required components, including a Root-of-Trust software layer, an operating system and connectivity, as well as additional hardware components as required by the use case.

The LPC55S1x MCU family expands the world's first general purpose Cortex-M33-based MCU series, offering significant advantages for developers, including pin-, software- and peripheral-compatibility for ease of use and to accelerate time to market, while leveraging the cost-effective 40-nm NVM process technology.

The LPC55S1x is the baseline family within the LPC5500 MCU series, providing new levels of cost and performance efficiency in addition to advanced security and system integration for industrial and general embedded markets.

### 1.5.1 Platform Security Features and Scope

The LPC55S1x offers the following security features:

- ARM TrustZone enabled.
- PRINCE module for real-time encryption of data being written to on-chip flash and decryption of encrypted flash data during read to allow asset protection
- CASPER Crypto co-processor is provided to enable hardware acceleration for various functions required for certain asymmetric cryptographic algorithms, such as Elliptic Curve Cryptography (ECC)
- AES-256 encryption/decryption engine
- Secure Hash Algorithm (SHA2) module supporting secure boot with dedicated DMA controller
- Physical Unclonable Function (PUF) using dedicated SRAM for silicon fingerprint. PUF can generate, store, and reconstruct key sizes from 64 to 4096 bits. Includes hardware for key extraction
- 128-bit unique device serial number for identification (UUID)
- Secure GPIO
- True Random Number Generator (TRNG)
- Code Watchdog

The functional block diagram is shown in the figure below. This diagram provides a view of the chip's major functional components and core complexes.

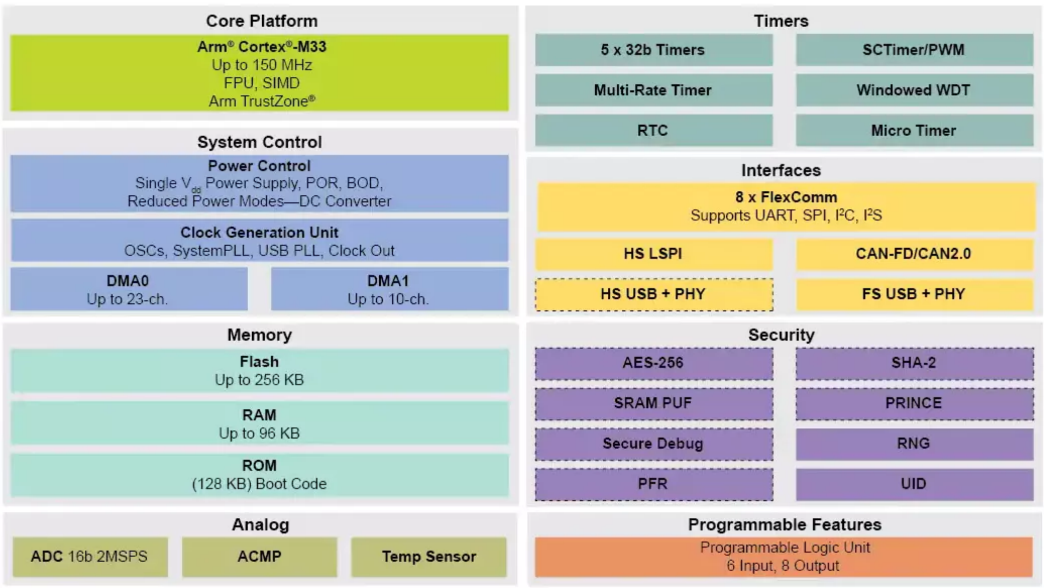


**Figure 1. Functional Block Diagram**

Figure 2 shows the security sub-system including Firmware functionality supported by the LPC55S1x.

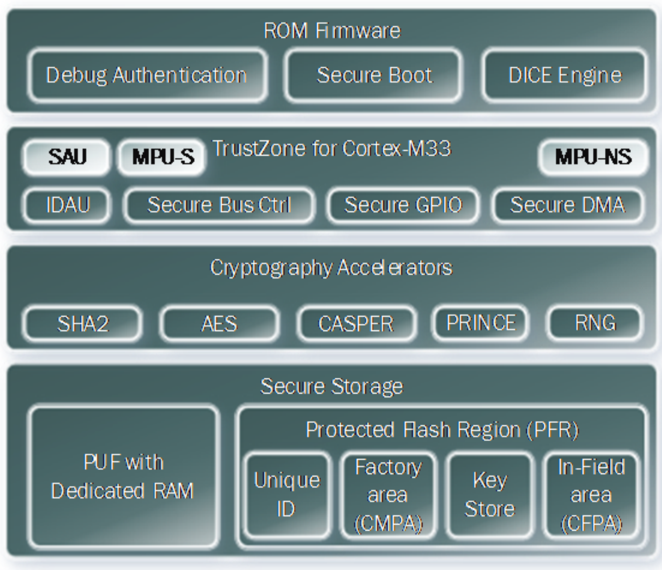


**Figure 2. Security sub-system**

The physical scope is the LPC55S1x microcontroller silicon chip including the on-chip ROM. The hardware components and interfaces are listed in Session 2 of [5].

The logical scope includes the ROM firmware, and the optional flash loadable crypto drivers as listed in Table 4. Any additional firmware, OS or application software stored on the platform is not in scope of this evaluation.

No additional non-platform hardware, software or firmware is required for the correct functioning of the security claims described in this document.

Table 4. Platform Deliverables

| Type | Name | Release | Form of delivery |
|---|---|---|---|
| IC Hardware | LPC55S1x | Rev. A0 | Silicon Chip |
| ROM Firmware | LPC551x/S1x ROM | 3.0.0 | On Chip ROM Firmware |
| ROM Firmware Patch | LPC551x/S1x ROM Patch | Rev 3 | On Chip Firmware |
| SDK (Optional) | LPCXpresso55S16 SDK | 2.8.0 | Software package |

### 1.5.2 Life Cycle

A reference of LPC55S1x MCU life cycle model can be found in Section 1.3 of [8]. The details on each of the states and transitioning between states are provided in Section 10 of [4].

### 1.5.3 Use Case Environments

**[trusted user only]**

The product may be operated in controlled environments. A controlled environment is typically enforced by strong organizational policies and means or by high trust in entities and users who handle the product.

**[any code]**

It cannot be excluded that the product executes code which is unknown to the product developer.

# 2 Security Objectives for the Operational Environment

## 2.1 Platform Objectives for the Operational Environment

For the platform to fulfill its security requirements, the operational environment (technical or procedural) must fulfill the following objectives:

- The operating system or application code are expected to verify the correct version of all platform components it depends on, by reading `SYSCON->DEVICE_ID0` and `SYSCON->DEVICE_ID1` (DIEID) for hardware identifier and revision number as specified in Tables 167 and 168 of [4], and using GetProperty command in ISP mode as described in Section 8.6 of [4] with tag value `0x01` to get ROM version as specified in Section 1.3, and with tag value `0x18` to get ROM patch revision in format T1.0.*x* where *x* is one plus the revision number specified in Section 1.3.
- The operating system or application code are expected to make use of the Secure Boot feature of signed image as described in Section 2.3 of [6] and Sections 6 and 7 of [4]. The operational environment is expected to ensure the security of the key(s) signing the secure boot image.
- The operating system or application code are expected to either disable debug by setting `DCFG_CC_SOCU_PIN` to `0x03FF` and `DCFG_CC_SOCU_DFLT` to `0`, or enable debug authentication by setting both `DCFG_CC_SOCU_PIN` and `DCFG_CC_SOCU_DFLT` to `0` as described in Table 1113 in Section 51 of [4]. The operational environment must ensure the security of debug authentication credentials, and no application/customer data is compromised by abusing authenticated debug functionality.
- The operational environment must protect the product against physical access of attackers as described in Section 1.5.3.
- The operating system or application code are expected to enable secure communication for security update, and in case of update, the update image is expected to be properly signed as described in Section 2.3 of [6], and distributed in secure manner to ensure confidentiality as well. The operating system or application code are expected to revoke an image as described in Section 2.3.1 of [6] in case of security incidence occurrence of the image.
- The operational environment is expected to provide secure OS and/or application code. To allow execution of unknown code while maintaining the protection of platform security features as declared in Section 3, the operating system or application code are expected to utilize the Cortex-M33 with full TEE and TrustZone enabled, including configuring restrictive memory boundaries via the MPU as described in Section 48 of [4].
- The operating system or application code are expected to configure the features and ensure secure and correct use of crypto and security service functionality as specified in Section 49 of [4].
- The operating system or application code are expected to ensure the correct version of the optional component is integrated, configure the features and ensure secure and correct use of crypto and security service functionality API as specified in associate SDK API and mbedTLS reference if the optional driver is used as specified in Chapter 4 of [10].
- The operating system or application code are expected to provide lifecycle states and secure mechanism of lifecycle state transition according to the use case, and the operational environment is expected to configure the platform accordingly for lifecycle state transitions as described in Section 10 of [4]. In general, the operating system or application code are expected to configure the platform to OEM closed state

no later than entering in-field application and Returned State (aka FA mode) before decommissioning or fault analysis.

- Before FA mode is activated, the operational environment is expected to purge information not protected by keys in KEYSTORE Flash page.
- The operational environment periodically monitors security update for the platform and react in a timely manner, by monitoring platform website, contacting NXP customer support, and/or updating from an accredited distributor.
- The operating system or application code are expected not to use SHA1 unless for backward compatibility reason and security impact and implication are analyzed and acceptable for the use case.

# 3 Security Requirements and Implementation

## 3.1 Security Assurance Requirements

The claimed assurance requirements package is: **SESIP2** as defined in Chapter 4 of GlobalPlatform Technology Security Evaluation Standard for IoT Platforms (SESIP), version 1.0 [1].

### 3.1.1 Flaw Reporting Procedures (ALC_FLR.2)

In accordance with the requirement for flaw reporting procedures (ALC_FLR.2), the developer has defined the following procedure:

NXP has defined a Product Security Incident Response Process (PSIRP), implemented by a dedicated team (PSIRT). This process provides a publicly available interface (https://nxp.com/psirt), and includes 4 steps:

- **Reporting**. The process begins when the PSIRT becomes aware of a potential security vulnerability in an NXP product. The reporter receives an acknowledgment and updates throughout the handling process.
- **Evaluation**. The PSIRT confirms the potential vulnerability, assesses the risk, determines the impact and assigns a processing priority. If the vulnerability is confirmed, the priority determines how the issue is handled throughout the remaining steps in the process.
- **Solution**. Working with PSIRT, the product team develops a solution that mitigates the reported security vulnerability. Solutions will take different forms based on the vulnerability. Because of the nature of NXP products – mostly silicon products where the firmware is in ROM -, very often the solution can only be provided in a next version of the chips and the short-term solution will consist of recommending security measures to be applied in systems using the NXP product.
- **Communication**. As said above, because of the nature of the NXP products, the solution to systems using the affected products often needs to be found in additional countermeasures in those systems. The communication on the vulnerability and solutions will in most cases be done directly towards the affected customers. For previously unknown or unreported issues, NXP will acknowledge the reporter of the issues (unless the reporter requests otherwise).

The firmware located in the on-chip ROM of the platform cannot be updated or patched. However, the platform’s Secure Boot feature is able to verify the authenticity of customer code during the initial boot and outside of the boot sequence, providing an appropriate mechanism for supporting the update of this code. The update mechanism itself has to be provided by the customer, most likely at the operating system level and is not in scope of this evaluation.

## 3.2 Base PP Security Functional Requirements

LPC55S1x fulfills the following security functional requirements:

### 3.2.1 Verification of Platform Identity

The platform provides a unique identification of the platform, including all its parts and their versions.

**Conformance rationale:**

Hardware identifier and revision number can be identified by reading `SYSCON->DEVICE_ID0` and `SYSCON->DEVICE_ID1` (DIEID) as specified in Tables 167 and 168 of [4], and using GetProperty command in ISP mode as described in Section 8.6 of [4] with tag value `0x01` to get ROM version, and with tag value `0x18` to get ROM patch revision in format T1.0.x where x is one plus the revision number.

### 3.2.2 Secure Initialization of Platform

The platform ensures its authenticity and integrity during the platform initialization. If the platform authenticity or integrity cannot be ensured, the platform will go to *locked state*.

**Conformance rationale:**

When the device boots, the execution starts in the device's physical ROM by the secure boot mechanism that verifies the authenticity of the firmware before executing it. The signature uses RSASSA-PKCS1-v1_5 signature [15] of a SHA256 digest with 2048-bit or 4096-bit key size as described in [6] and Sections 6 and 7 of [4].

### 3.2.3 Secure Debugging

The platform only provides *Arm's Serial Wire Debug (SWD) interface* authenticated as specified in Section 51.7 of [4] with debug functionality.

The platform ensures that all data stored by the application, with the exception of *all data*, is made unavailable.

**Conformance rationale:**

The platform offers a debug authentication protocol as a mechanism to authenticate the debugger (an external entity) has the credentials approved by the product manufacturer before granting debug access to the device. The debug authentication scheme on the platform is a challenge-response scheme based on 2048- or 4096-bit RSASSA-PKCS1-v1_5 signature verification, and assures that debugger in possession of required debug credentials only can successfully authenticate over debug interface and access restricted parts of the device.

Note that once debug authentication succeeds, the debugger has full access of the debug domains that are not permantantly disabled, therefore, procedures are needed to ensure the debug session is not abused.

### 3.2.4 Secure Update of Platform

The platform can be updated to a newer version in the field such that the integrity, authenticity and confidentiality of the platform is maintained.

**Conformance rationale:**

The hardware component and ROM firmware are immutable, therefore cannot be updated. Please also refer to Section 3.1.1 for flaw reporting procedures.

The Secure Boot ROM also provides supports of secure firmware update so that with proper implementation and configuration, the final product can update in a secure manner. As described in Section 3.2.2, RSASSA-PKCS1-v1_5 signature verification during secure boot also applies for an updated firmware.

The ROM further supports public keys and image revocation i.e. the method of not allowing new updates to be applied unless they are of a specific version. This is the basis for roll back protection.

### 3.2.5 Residual Information Purging

The platform ensures that *all KEYS and IVs in KEYSTORE Flash page*, with the exception of *data other than KEYSTORE Flash page*, is erased using the method specified in *Section 51 of* [4] before the memory is (re)used by the platform or application again and before an attacker can access it.

**Conformance rationale:**

Platform ROM offers the FA Mode (SET_FA_MODE) command handler to enable deletion of sensitive information (for example, Keys). The ROM allows the SET_FA_MODE command only when corresponding flag in 'debug_state' is set.

Activation of the FA_MODE boot sequence will perform the following:

- Create a new version of Customer Field Programmable (CFPA) page.
- Set ENABLE_FA_MODE word in the page to value 0xC33CA55A.
- Erase all KEYS and IVs in KEYSTORE Flash page.
- Flush all temporary key registers.
- Blocks PUF indexes.
- Open all debug ports.
- Enter a while (1) loop.

Note that the information encrypted by keys in KEYSTORE flash page will be purged equivalently as keys are purged when FA_MODE is activated.

Also, as debug ports are open after entering FA_mode, the Operational Environment is expected to purge information not protected by keys in KEYSTORE Flash page.

## 3.3 Package “Security Services” Security Functional Requirements

### 3.3.1 Cryptographic Operation

The platform provides the application with *encryption and decryption* functionality as specified in *FIPS 197 (AES)* [13] for key length *128, 192 or 256 bits* and modes *ECB and CBC and CTR*.

The platform provides the application with *hashing* functionality as specified in *FIPS 180-4* [12] for digests of *160 bits (SHA-1) and 256 bits (SHA-256)*.

The platform provides the application with *signature generation and verification* functionality with ECDSA as specified in *ISO/IEC 14888-3:2015* [16] for key length *256 bits* and modes *not applicable*.

The platform provides the application with *Diffie-Hellman* functionality with ECDH as specified in *NIST FIPS 800-56A* [14] for key length *256 bits* and modes *not applicable.*

**Conformance rationale:**

The support for cryptographic operations of AES and SHA is described in Section 49 of [4]. SHA1 is supported for backward compatibility only. Refer to Section 2.1.

CASPER crypto co-processor is provided to enable hardware acceleration for various functions required for asymmetric cryptographic algorithms, such as Elliptic Curve Cryptography (ECC) as described in Section 50 of [4] and [7]

Supported cryptographic functions are implemented in the SDK (Software Development Kit) and the mbed TLS examples utilize the CASPER peripheral for computations.

### 3.3.2 Cryptographic Key Generation

The platform provides the application with a way to generate cryptographic keys for use in *ECC* as specified in [14] and [16] for key lengths *256 bits*.

**Conformance rationale:**

ECC key generation examples are available in mbed TLS in SDK.

### 3.3.3 Cryptographic KeyStore

The platform provides the application with a way to store *cryptographic keys* such that not even the application can compromise the *authenticity, integrity, confidentiality* of this data. This data can be used for the cryptographic operations *encryption, decryption, signature generation*.

**Conformance rationale:**

The TOE supports hardware unique keys, managed by the PUF KeyStore, which include a 256-bit AES key and three 128-bit PRINCE keys derived from a PUF output; these keys are never accessible in main memory, as it is directly fed to the AES or PRINCE accelerator when needed.

For the authentication checks during boot, several keys can be used to sign the files. A hash of the hashes of the corresponding public keys is stored on the chip's Protected Flash Region (PFR) and is used to verify the validity of the public key in the boot image.

Other secret keys used by the secure processing environment can be derived from the hardware unique key and can be managed directly by the PUF KeyStore.

### 3.3.4 Cryptographic Random Number Generation

The platform provides the application with a way based on *physical noise* to generate random numbers to as specified in AIS31 (P1/PTG.1) [3].

**Conformance rationale:**

The True Random Number Generator (TRNG) on the platform is based on two main sources of entropy:

• Phase noise of unprecise clocks derived from the ring oscillators.

• The default values of hundreds of internal flip-flops after a reset.

## 3.4 Package "Software Isolation" Security Functional requirements

### 3.4.1 Software Attacker Resistance: Isolation of Platform

The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise any other claimed security functional requirements.

**Conformance rationale:**

TrustZone provides the means to implement separation and access control to isolate trusted software and resources to reduce the attack surface of critical components. The trusted firmware can protect trusted operations and is ideal to store and run the critical security services. The code protects trusted hardware to augment and fortify the trusted software. This includes the modules for hardware assists for cryptographic accelerators, random number generators, and secure storage. Best practices demand that that this code be small, well-reviewed code with provisions of security services.

The platform has implemented Cortex-M33 with full TEE and TrustZone support enabled.

## 3.5 Additional Security Functional Requirements

### 3.5.1 Verification of Platform Instance Identity

The platform provides a unique identification of that specific instantiation of the platform, including all its parts and their versions.

**Conformance rationale:**

The platform stores a 128-bit IETF RFC4122 compliant non-sequential Universally Unique Identifier (UUID). It can be read from the flash PFR region at register location 0x0009_FC70 onwards. For version information, refer to Section 3.2.1.

### 3.5.2 Software Attacker Resistance: Isolation of Platform Parts

The platform provides isolation between platform parts, such that an attacker able to run code in *Non-Secure* can compromise neither the integrity and confidentiality of *Secure* nor the provision of any other security functional requirements.

**Conformance rationale:**

The platform has implemented Cortex-M33 with full TEE and TrustZone support enabled. Refer to Section 3.4.1 for more information.

### 3.5.3 Secure Encrypted Storage

The platform ensures that all data stored by the application, except for *data not stored in the configured address area*, is encrypted as specified in [11] with a platform instance unique key of key length *128 bits*.

**Conformance rationale:**

The platform offers support for real-time encryption and decryption for on-chip flash using the PRINCE encryption algorithm [11]. Compared to AES, PRINCE is fast because it can decrypt and encrypt without adding extra latency. PRINCE operates as data is read or written, without the need to first store data in RAM and then encrypt or decrypt to another space. It operates on a block size of 64-bits with a 128-bit key. This functionality is useful for asset protection, such as securing application code, securing stored keys, and enabling secure flash update.

The platform supports three regions for encryption and decryption, referred to as crypto regions. Each crypto region resides at a 256 kB address boundary within the flash. All three regions have a start address of 0x0 and all three regions are overlapped. Each crypto region is divided into 8 kB sub-regions which can be individually enabled.

Each crypto region has a dedicated key and IV. It allows multiple images to reside in the flash with an independent encryption base. The key is sourced from PUF via an internal hardware interface, without exposing it on the system bus.

# 4 Mapping and Sufficiency Rationales

## 4.1 SESIP2 Sufficiency

| Assurance Class | Assurance Family | Covered By | Rationale |
|---|---|---|---|
| ASE: Security target evaluation | ASE_INT.1 ST Introduction | Section 1 | The ST reference is in Section 1.1, the TOE reference in Section 1.3, the TOE overview and description in Section 1.5. |
| | ASE_OBJ.1 Security requirements for the operational environment | Section 2 | The objectives for the operational environment in Section 2 refer to the guidance documents. |
| | ASE_REQ.3 Listed security requirements | Section 3 | All SFRs in this ST are taken from[1]. SFR "Identification of Platform Type" is included. SFR "Secure Update of Platform" is mentioned but refers to ALC_FLR.2. |
| | ASE_TSS.1 TOE Summary Specification | Section 3 | All SFRs are listed per definition, and for each SFR the implementation and verification are defined in the SFR. |
| ADV: Development | ADV_FSP.4 Complete functional specifications | Section 1.4 and material provided to evaluator. | The evaluator will determine whether the provided evidence is suitable to meet the requirement. |
| AGD: Guidance documents | AGD_OPE.1 Operational user guidance | Section 1.4 | The evaluator will determine whether the provided evidence is suitable to meet the requirement. |
| | AGD_PRE.1 Preparative procedures | Section 1.4 | The evaluator will determine whether the provided evidence is suitable to meet the requirement. |
| ALC: Life-cycle support | ALC_FLR.2 Flaw reporting procedures | Section 3.1.1 | The flaw reporting and remediation procedure is described. |
| ATE: Test | ATE_IND.1 Independent testing: conformance | Material provided to evaluator. | The evaluator will determine whether the provided evidence is suitable to meet the requirement. |

| Assurance Class | Assurance Family | Covered By | Rationale |
|---|---|---|---|
| AVA: Vulnerability assessment | AVA_VAN.2 Vulnerability analysis | N.A.<br>A vulnerability analysis is performed by the evaluator to ascertain the presence of potential vulnerabilities. | The evaluator performs penetration testing, to confirm that the potential vulnerabilities cannot be exploited in the operational environment for the TOE. Penetration testing is performed by the evaluator assuming an attack potential of Basic. |

# 5 Bibliography

## 5.1 Evaluation Documents

[1] GlobalPlatform Technology Security Evaluation Standard for IoT Platforms (SESIP), version 1.0, GP_FST_070.

[2] Protection Profile for Secure MCUs and MPUs, V0.107, NXP Semiconductors.

[3] A proposal for: Functionality classes for random number generators, Wolfgang Killmann, T-Systems GEI GmbH, Werner Schindler, Bundesamt für Sicherheit in der Informationstechnik (BSI), Version 2.0, 18 September 2011.

## 5.2 Developer Documents

[4] UM11295, LPC55S1x/LPC551x User Manual, Rev.1.3, NXP Semiconductors.

[5] LPC55S1x/LPC551x Product data sheet, Rev.0.7, NXP Semiconductors.

[6] AN12283, LPC55Sxx Secure Boot, Rev. 1, NXP Semiconductors.

[7] AN12445, Asymmetric Cryptographic Accelerator CASPER, Rev. 3, NXP Semiconductors.

[8] AN12278, LPC55S00 Security Solutions for IoT, Rev. 1, NXP Semiconductors.

[9] AN12326, Secure GPIO and Usage, Rev. 1, NXP Semiconductors.

[10] MCUXpresso SDK API Reference Manual Rev. 0 in LPC55S16 SDK 2.8.0.

## 5.3 Standards

[11] J. Borghoff, et al, PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications, Cryptology ePrint Archive, Report 2012/529.

[12] FIPS PUB 180-4: Secure Hash Standard (SHS), Federal Information Processing Standards Publication, Information Technology Laboratory, National Institute of Standards and Technology, August 2015.

[13] FIPS PUB 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/National Institute of Standards and Technology, 26 November 2001.

[14] NIST Special Publication 800-56A Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography.

[15] PKCS #1: RSA Cryptography Standard, Version 2.2, October 27, 2012, RSA Laboratories

[16] ISO/IEC 14888-3:2015: Information technology â€“ Security techniques â€“ Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms, 2016.

# 6 Legal information

## 6.1 Definitions

**Draft** — The document is a draft version only. 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 herein 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.

**Limiting values** — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

**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.

**No offer to sell or license** — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

**Quick reference data** — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

**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.

**Non-automotive qualified products** — Unless this data sheet 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 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 or documented vulnerabilities. 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 react appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and makes 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.

## 6.3 Trademarks

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

# Tables

# Figures

# Contents





**Date of release: 15 September 2020**