AN12514

SE050 - User Guidelines

Rev. 1.4 — 21 November 2021 Application note

Document information

Information	Content
Keywords	Secure Element, SE050, User Guidelines, Plug & Trust, EdgeLock SE050
Abstract	This document provides the guidelines for the usability of SE050 and the security recommendations for using the module

Revision history

Rev	Date	Description
1.4	2021-11-21	Add a note in Section 1
1.3	2021-05-10	• Updated Section 8.1.1.1 • Updated Section 8.1.1.2 • Updated Section 8.1.1.3.
1.2	2020-12-15	Updated legal information
1.1	2020-06-22	• Updated Section 4.5.4 • Updated Section 4.5.5
1.0	2019-12-16	Initial version

1 Introduction

This document provides functional and security recommendations for the EdgeLock SE050 security module to system integrators and application developers.

A difference will be made between single-tenant use and multi-tenant use:

- **Single-tenant** means the SE050 does not protect credentials separately for different users. No separation of users on SE050 is needed to use any of the credentials.
- **Multi-tenant** means the SE050 separates access to credentials based on a secret (= authentication object). Multi-tenant can be multiple (physical) users, but also multiple different applications or even multiple threads in the same application.

The guidelines in this document for single-tenant are always applicable, both for single-tenant and for multi-tenant use of the SE050. Guidelines specific to multi-tenant use do not apply to single-tenant use.

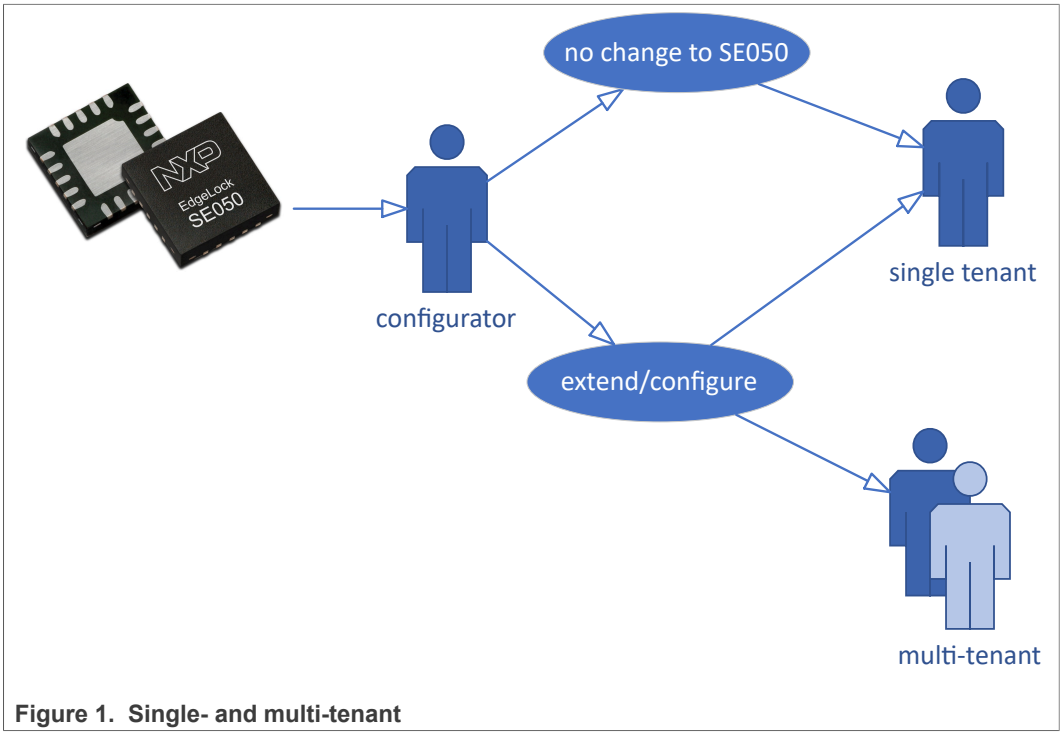

Figure 1. Single- and multi-tenant

The different chapters describe the different possible usages of the SE050:

- **SE050 basics**
  - Describes the basics of the SE050: what are Secure Objects and how they can be used.

- **SE050 Plug and Trust: Usage out of the box** (= use as-is as single-tenant)
  - describes how to use the SE050 secure element straight out of the box, with one single entity using the SE050. It provides general guidelines on the use cases which are feasible with the product variants off the shelf.
  - **Ease of Use Configuration** describes the generic SE050 variants which will be available in the market ready to plug and trust. These devices come with a specific set of credentials being trust-provisioned by NXP.
  - **Single-tenant Use Cases** provides an explanation of the most simple use cases applicable to the single-tenant usage of SE050. In this context single-tenant means that the SE050 is operated in a single instance.
- **SE050 Plug and Trust: extendibility** (= configure the SE050)
  - provides support to those users or system integrators who extend the Ease of Use Configuration with new provisioned credentials. This can apply for single-tenant use (e.g. adding additional credentials besides the Ease Of Use Configuration) or multi-tenant use (e.g. configure the SE050 for use by 2 different end users).
- **SE050 Plug and Trust: multi-tenant usage**
  - describes how to use the SE050 secure element by multiple entities. It provides general guidelines on the use cases which are feasible with the turnkey product variants.
  - **Module Description Advanced**
    - **Secure objects Advanced**
    - **Policies**
    - **Object Deletion**
  - **Trust Provisioning**
  - **Multi-tenant System**
    - **Authenticated Key Creation**
    - **Multi level SCP**
  - **Security Recommendations**
    - **Functional Recommendations**

Diagrams for each use case are present at the end of the chapter.

Note: *A newer version (1.5) of this user guidance manual is available on Docstore that applies to SE050A/B/C/D types only. Please contact your NXP representative to get access to the version 1.5.*

2 SE050 basics

This chapter explains some basics about the SE050 so users can start using the SE050. It repeats definitions and concepts in a short form as described in the APDU Specification, see: [1].

2.1 Unauthenticated user

For any single-tenant use case, the user can use the SE050 functionality without authentication when both conditions are met:

- there is no interaction between (multiple) users
- no access control is needed to protect credentials against other users.

2.2 Platform SCP

By default, any delivered SE050 device has a SCP03 base key set that contains the same keys for each device-type (non-die-individual, keys for device types specified in Application Note SE050 Configurations, see: [2]).

Users who want to protect the communication between a host processor and the secure element can use SCP03 on platform level. This secure channel including the key management to update the base keys can be fully managed by GlobalPlatform commands and does not need any SE050 specific APDU.

2.3 Unbound user

Regardless of the authentication on platform level, the user will not apply any authentication to the applet. This is referred to as an ***unbound*** user. Section "Sessions" will detail *bound* users.

2.4 Secure Objects

Anything that is stored or generated inside the SE050 is a Secure Object.

2.4.1 Secure Object types

Supported Secure Object types are:

- **Keys**
  - ECKey = asymmetric key on any of the supported elliptic curves
  - RSAKey = asymmetric key for RSA (raw or CRT format) of 512, 1024, 1152, 2048, 3078 or 4096 bit
  - AESKey = symmetric key of 128, 192 or 256 bit; used for AES cipher operations
  - DESKey = symmetric keys used for DES cipher operations
  - HMACKey = symmetric key of any bit length; used for HMAC and HKDF operations.
- **Files**
  - BinaryFile = a byte array (i.e. general purpose storage)
  - Counter = a monotonic counter
  - PCR = a hash value that can be extended with extra data
  - UserID = a byte array that can be used to group secure objects and allow their usage in an associated session (intended for use cases where a trusted operating system on a host MCU/MPU is isolating their applications based e.g. on their application ID).

See the [1] for more information.

2.4.2 Secure Object Attributes

Secure Object attributes are linked to any Secure Object. The attributes are:

- *Object identifier* = a unique identifier of the Secure Object
- *Type* = Secure Object type
- *Policy* = Access control applicable to the secure object
- *Origin* = Origin of the data, either external, internal or provisioned

- Additional attributes (only applies to multi-tenant; see Multi-tenant use of SE050)
  - *Authentication attribute*
  - *Object counter*
  - *Authentication object identifier*
  - *Maximum authentication attempts*

2.4.2.1 Object identifier

The object identifier cannot be modified during the object lifetime, so it remains the same until the object is deleted.

Object identifiers are always defined externally, the SE050 will not (automatically) assign object identifiers to objects. However, there is a set of reserved identifiers that are assigned to serve specific use cases. For more information, see the [1].

All pre-provisioned credentials being trust-provisioned by NXP as part of the Ease Of Use Configuration have an identifier from the range “Applet Reserved Area” or “NXP reserved region” in Table 1. Customers that will create their own Secure Objects are advised to use an identifier from the range “In field usage”.

Table 1. Identifier for Applet reserved area or NXP reserved region

Address Range	IDs
0x00000000-0x7BFFFFFF	In field usage
0x7C000000-0x7CFFFFFF	Android Key Master area
0x7D000000-0x7DFFFFFF	Demo area
0x7FFF0000-0x7FFFFFFF	Applet reserved area
0x80000000-0xFFFFFFFF	NXP reserved region

2.4.2.2 Type

See Secure object types.

2.4.2.3 Policy

A Secure Object policy defines the access control to a Secure Object by specifying the operations that each user can perform (note that for single-tenant use, the unbound user applies).

Secure Object policies are assigned at object creation and cannot change afterwards, so they remain constant over the lifetime of a Secure Object.

If no policy is passed to a Secure Object at object creation, a default policy will apply.

The default policy is shown in Table 2. It allows all supported operations, except attestation. For an explanation on the purpose of a specific policy, see the [1].

Table 2. Default policy for Secure Objects

'V' means operation supported, '-' means operation not supported, 'X' means operation supported, but prohibited by the default policy.

Policy	AESKey DESKey HMACKey	RSAKey ECKey	BinaryFile Counter PCR	UserID
POLICY_OBJ_ALLOW_DELETE	V	V	V	V
POLICY_OBJ_REQUIRE_SM	V	V	V	V
POLICY_OBJ_REQUIRE_PCR_VALUE	V	V	V	V
POLICY_OBJ_FORBID_ALL	V	V	V	V
POLICY_OBJ_ALLOW_SIGN	V	V	-	-
POLICY_OBJ_ALLOW_VERIFY	V	V	-	-
POLICY_OBJ_ALLOW_ENC	V	V	-	-
POLICY_OBJ_ALLOW_DEC	V	V	-	-
POLICY_OBJ_ALLOW_KDF	V	V	-	-
POLICY_OBJ_ALLOW_WRAP	V	V	-	-
POLICY_OBJ_ALLOW_WRITE	V	V	V	V
POLICY_OBJ_ALLOW_GEN	V	V	-	-
POLICY_OBJ_ALLOW_KA	-	V	-	-
POLICY_OBJ_ALLOW_READ	-	V	V	-
POLICY_OBJ_ALLOW_ATTESTATION	-	X	-	-
POLICY_OBJ_ALLOW_DESFIRE_AUTHENTICATION	V	-	-	-
POLICY_OBJ_ALLOW_DESFIRE_DUMP_SESSION_KEYS	V	-	-	-
POLICY_OBJ_ALLOW_IMPORT_EXPORT	V	V	-	-

2.4.2.4 Origin

The origin attribute indicates the origin of the content of a Secure Object: either externally generated, internally generated or if trust provisioned by NXP.

2.4.3 Product identification

Product variant can be checked with Get Version Command. Expected answers to be found in the SE050 configuration AN, see: [2].

The Product Variant is exactly identifiable using the NXP OEF ID. The list of existing standard product variants in listed in [2]

The example "get info" which is included in the Plug&Trust MW as source and precompiled binary prints out this variant identifier, e.g. for the Demoboard: App:INFO :OEF ID (Len=2) A1 F4as well as the active product features, e.g. on the demoboard:

```
App:INFO :OEF ID (Len=2)
                  A1 F4
```

as well as the active product features, e.g. on the demoboard:

```
App:INFO :Applet Major = 3
App:INFO :Applet Minor = 1
App:INFO :Applet patch = 0
App:INFO :AppletConfig = 6FFF
App:INFO :With      ECDAA
App:INFO :With      ECDSA_ECDH_ECDHE
App:INFO :With      EDDSA
App:INFO :Wit       DH_MONT
App:INFO :With      HMAC
App:INFO :With      RSA_PLAIN
App:INFO :With      RSA_CRT
App:INFO :With      AES
App:INFO :With      DES
App:INFO :With      PBKDF
App:INFO :With      TLS
App:INFO :With      MIFARE
App:INFO :With      I2CM
```

3 SE050 Plug and Trust: Usage out of the box

3.1 Ease of Use Configuration

All generic SE050 variants will be offered pre-provisioned according to a profile.

The variants available are: A, B, C.

For more details, refer to [2].

3.2 Single-tenant protection

The configurator is responsible to bring the SE050 in a usable state for the end user's application.

The configurator and end user might be the same entity, e.g. a developer for prototyping purpose.

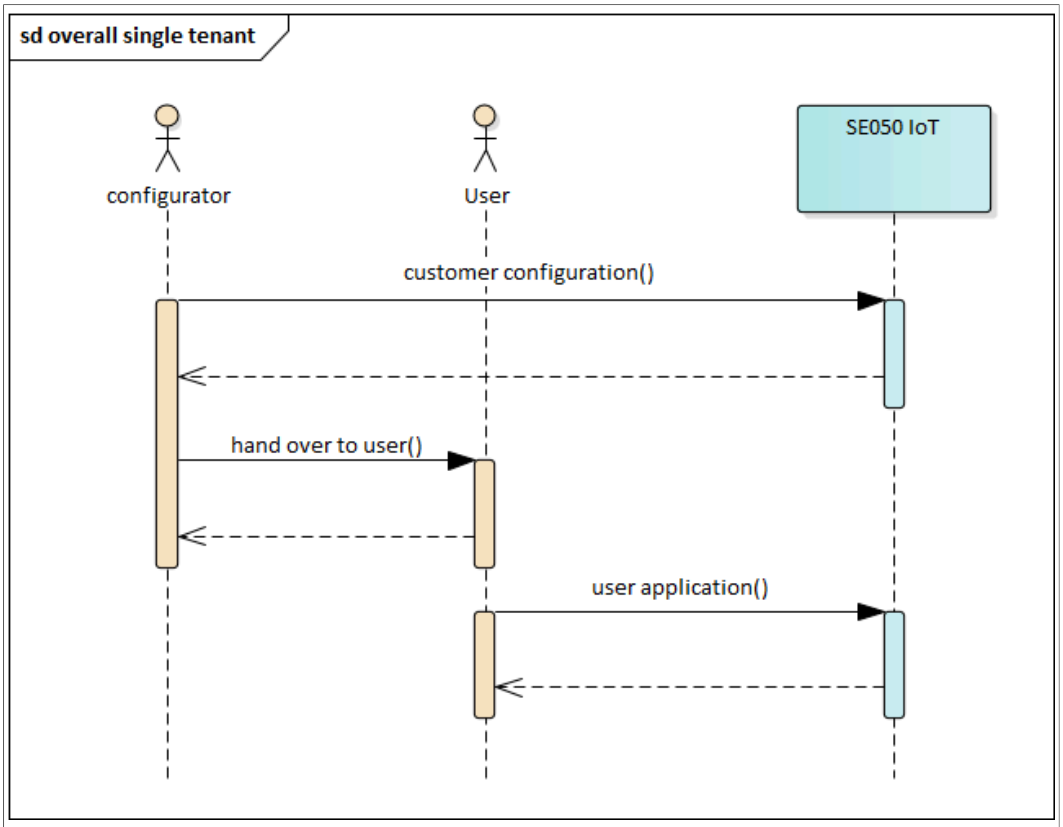

Figure 2. Overview single-tenant use

Note:

Configurator and User can be the same entity

3.2.1 Update Platform SCP keys

SCP stands for Secure Channel Protocol. SE050 uses the standard Global Platform secure channel APDUs.

To protect from local attack on the physical interface, the usage of PlatformSCP is recommended. This allows to securely bind the Host MCU and the secure element by using pre-shared symmetric keys to establish an e2e encrypted and integrity protected communication channel.

The Platform SCP implements SCP03 protocol. SCP03 requires a set of 3 AES128 master keys. These keys need to be also stored into the host MCU to pair it with the SE050 Module.

Security recommendations on SCP are described in Platform SCP.

3.2.1.1 How to update Platform SCP keys

The Platform default keys are available in [2].

SCP03 base keys can be updated as described in GlobalPlatform [GPC_2_2_D-SCP03v1.1_c] Section 7.2, using the existing DEK key as encryption key for the new keys to be set.

The Plug&Trust Middleware contains an example program to rotate (i.e. update) the Platform SCP03 keys.

The example program – and additional documentation - can be found in the demo folder.

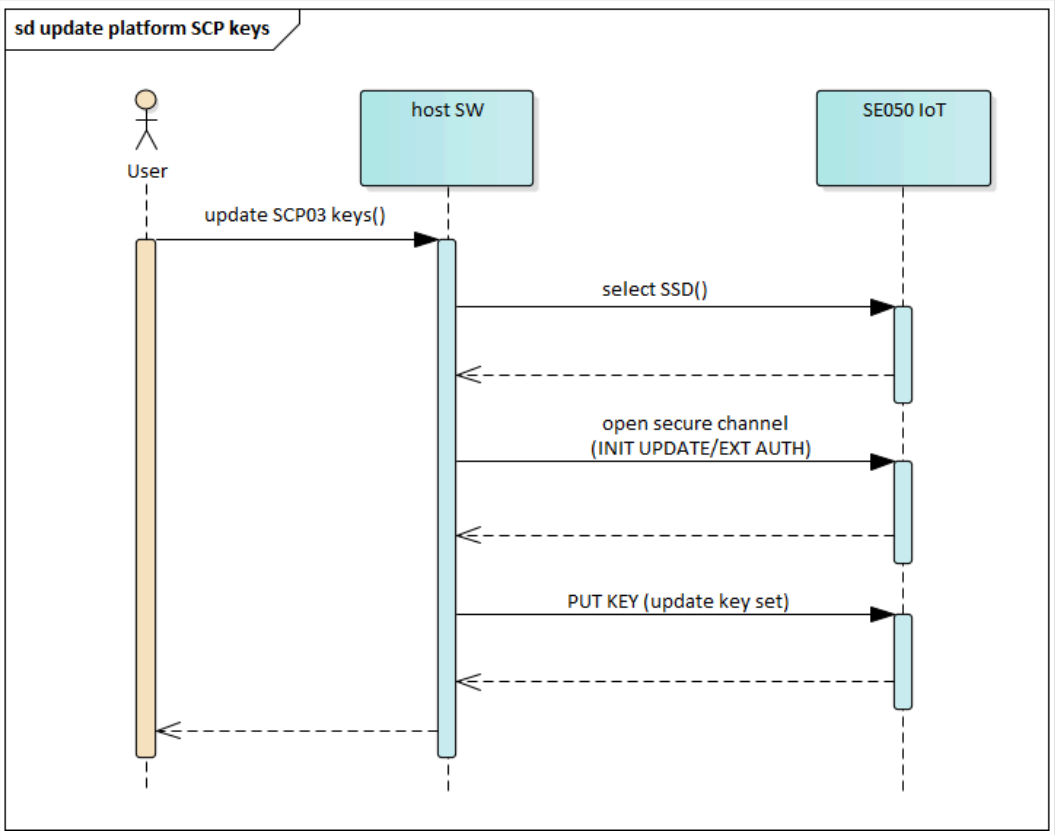

Figure 3. Enable Platform SCP

3.2.2 Attestation

Attestation is a means to prove that the data is originated in the secure element. The secure element “attests” the origin of the data by signing it with a trust provisioned attestation key by NXP. When key or file data are requested by the user, the user can request attestation for the returned data. Attestation is achieved by adding into the response of the requested data the chip unique identifier + freshness (i.e. a random value) + a timestamp (i.e. monotonic counter value) + a signature over the full payload (requested data + unique identifier + freshness + timestamp).

All the generic SE050 variants have an attestation key trust provisioned by NXP. Variant C also contains an NXP signed certificate associated to the attestation key.

The certificate is signed by NXP Root of Trust entity. Attestation requires trust which is ensured by the issued certificate. To verify the validity of the attestation, the signature on the attested object is checked against the attestation certificate.

Security recommendations on attestation are detailed in Attestation”.

Use cases

1. Generated Key attestation
   SE050 can generate keys internally. The attestation mechanism is used to attest that the keys have been generated inside the SE050.
2. External Data attestation
   Customer might inject/provision data inside the SE050. The attestation mechanism can be used to prove that the data has been stored in the secure element without modification.

For more information, see Attestation of provisioned objects.

4 SE050 configuration extendibility

This chapter provides information to those users and customers who desire to extend the SE050 product variants with customized keys and credentials besides the Ease Of Use configuration.

Security recommendations for extending SE050 Ease of Use configuration are provided in Extendibility and Multi-tenant.

4.1 Adding Secure Objects

Users can add Secure Objects to the Ease of Use configurations by creating new Secure Objects. During creation, the user needs to assign the object identifier (see Object identifier). A policy can be set if the default policy is not sufficient. For access control it is recommended to set a policy on Secure Objects as stated in section Secure Object policy.

Users have to choose between persistent and transient Secure Objects (in case both are supported; see [1]). Persistent Secure Objects value is always written to NVM while transient Secure Objects value is written into RAM.

For transient Secure Objects, it is possible to export the Secure Object to the host controller and later on import the Secure Object again. Persistent Secure Objects cannot be exported or imported.

4.2 Creating Crypto Objects

By default, users have the possibility to do either cipher, signature or digest operations in one shot, meaning the input data are passed to the SE050 and the output data are returned directly. The SE050 does not keep any state in that case.

However, when users have the need to pass several blocks consecutively to the SE050 for one of those cipher, signature or digest operations, a Crypto Object can be allocated.

A Crypto Object will keep the state of a crypto operation and allow typically to do init/(n times) update/final operations.

4.3 Adding an attestation key

Customers can provide their own attestation key (and related certificates) to perform the attestation as explained in Attestation. The key which is used for attestation MUST have the POLICY_OBJ_ALLOW_ATTESTATION explicitly set, the default policy does not grant the attestation right.

4.4 Adding Cloud Connection keys

SE050 Ease of Use configuration can be extended with credentials to onboard and connect securely to various clouds. Customers might decide to use their own PKI and CA.

More details on this use case can be found in [3] and [4].

4.5 Apply transport lock

Transport lock use provides security recommendations to securely use the transport lock.

4.5.1 Simple Use Case

The transport lock is a secure object which can be used to protect the modules on the logistic chain.

The transport lock MAY be used as tamper seal between entity A and entity B. Entity A

applies a lock and share the key with entity B to give authorized access only to entity B. In this case entity B MAY be the final receiver of the devices

4.5.2 Updatable Transport Lock

There might be more than two entities in the logistic chain. In this scenario each entity MAY be both a customer and a configurator.

In the case of a cascade logistic chain, entity A MAY make the Transport Lock updatable.

Receiving entity B can remove the lock of A and update the lock for further entities into the logistic chain.

Entity B MAY apply a specific lock for each entity which will receive the product.

4.5.3 Factory reset

Factory key reset allows to delete all objects except for those where the origin attribute is configured to "provisioned". This is the case for all keys belonging to the NXP Ease of Use configuration as well as to mandatory secure objects such as UUID.

Note: Certificates trust provisioned by NXP will be deleted after factory reset.

4.5.4 Object deletion

As described above, some of the credentials injected into the Ease of Use configuration, such as the Cloud Onboarding certificates, can be deleted by the customer if desired.

To delete them, the credentials must be overwritten first. This will change the origin from “provisioned” to either “internal” or “external” (depending on the write method) and will allow deletion either via individual deletion or via the factory reset.

Note: *To ensure the correct execution of the deleteAll command, an attested read on a previously existing object must be performed after the execution of the deleteAll command. The attested read response must indicate that the object has been correctly deleted.*

4.5.5 Importing external objects

Note: *The APDU "ImportExternalObject" must not be used without first contacting NXP to avoid potential problems. If you have used or plan to use the APDU "ImportExternalObject," please make sure you contact your NXP representative.*

Users might import credentials using importExternalObject or writing to a new object using either Applet-SCP03 or FastSCP session.

ImportExternalObject is by default only possible using an NXP service as the necessary key (ID 0x7FFF0202, RESERVED_ID_KP_FASTSCP_IMPORT) is provisioned already by NXP as specified in [2].

Any command that is used to write or generate a credential can be sent, but in addition to a normal write/generate operation, the command itself is signed and thus implicitly authenticated before being executed.

4.6 Single-tenant use cases

4.6.1 Cloud Connection

The SE050 Ease of Use configuration can be used to onboard and connect securely to various clouds. More details can be found on the following Application Notes:

- SE050 Secure Connection OEM Cloud Application Note
- SE050 Secure Connection Google Cloud IoT Core Application Note
- SE050 Secure Connection Watson IoT Device to Device Authentication Application Note

4.6.2 Device to Device Authentication

Every SE050 product variant comes pre-provisioned with credentials which can be used for Device to Device authentication.

More details on the use case can be found in [8].

4.6.3 Attestation of provisioned objects

Note that for attestation, the key pair that performs the attestation (i.e. signing) needs to be part of a trusted certificate chain.

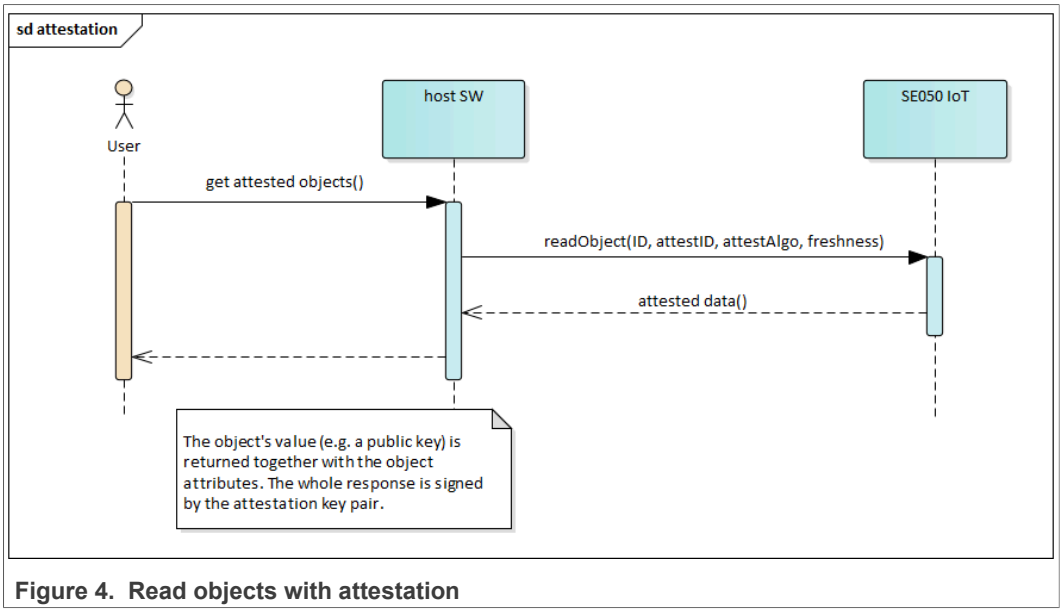

Figure 4. Read objects with attestation

4.6.4 User application

Any command can be sent in the default session (no authentication to the applet nor command wrapping are needed).

Data are encrypted and protected by Platform SCP.

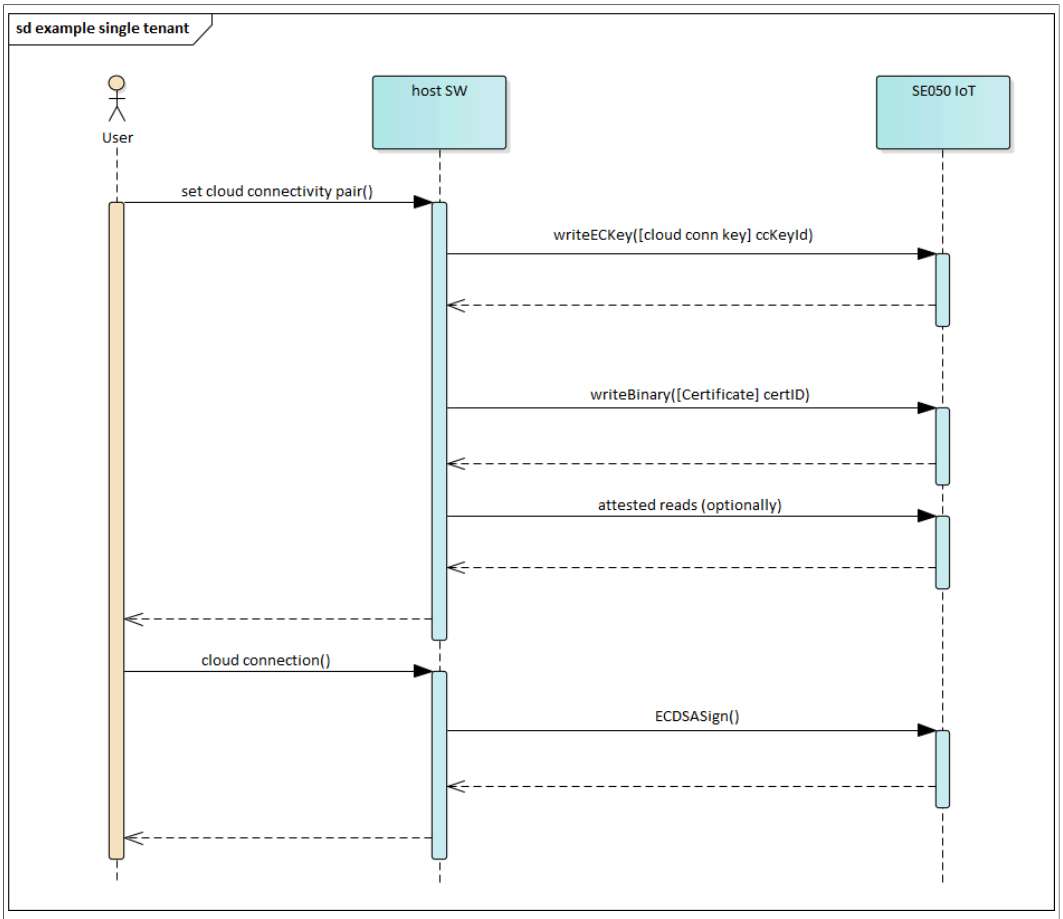

Figure 5. Single-tenant user application example

5 Multi-tenant use of SE050

5.1 SE050 features for multi-tenant use

5.1.1 Authentication Objects

An Authentication Object is a specific Secure Object that allows users to mutually authenticate against the SE050 applet. In that sense the value of the object is protecting access to the SE050.

Users who use an Authentication Object to authenticate against the SE050 applet are referred to as ***bound*** users (as opposed to the *unbound* user).

5.1.1.1 Authentication Object Creation

The entity that creates the Authentication Object is referred to as the ***authentication object owner***. The owner can be a single entity or multiple entities (whoever knows the value of the Authentication Object's value).

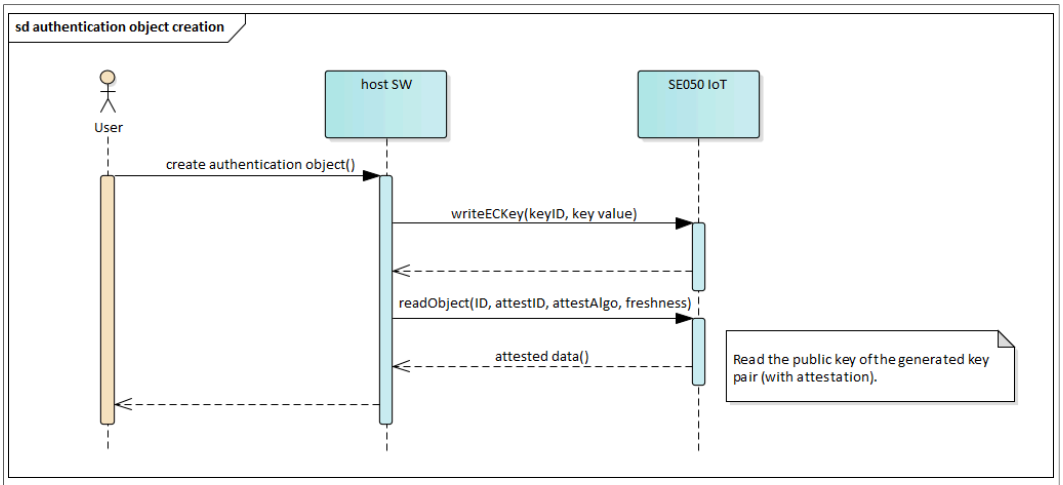

Figure 6. Authentication object creation (example: ECKey pair)

5.1.2 Sessions

The SE050 allows to open a **session** with any of the following Secure Objects:

- userID (to open a UserID session)
- AES128 key (to open a SCP03 session)
- ECKey pair or EC public key (to open a FastSCP session)

UserID sessions are using communication in plain, whilst SCP03 or FastSCP sessions rely upon SCP03 secure messenging and therefore provide end-to-end protection (see: security recommendation.

If a user opens a session:

- the access rights for the unbound user do not longer apply
- the user will become known by the Authentication Object ID that was used to open the session and becomes a ***bound*** user.

By default, if a SCP03 or FastSCP session is established, **applet level SCP** will apply end-to-end for this particular session. Applet level SCP is using SCP03 secure messaging; the difference between an SCP03 and a FastSCP session is the authentication method, where SCP03 is based on symmetric crypto and FastSCP is using asymmetric crypto for the authentication.

Authenticated User Session provides recommendations for a secure use of sessions.

Note that in cases where Platform SCP is used, the applet level SCP is wrapped inside of the Platform SCP channel, see: [1].

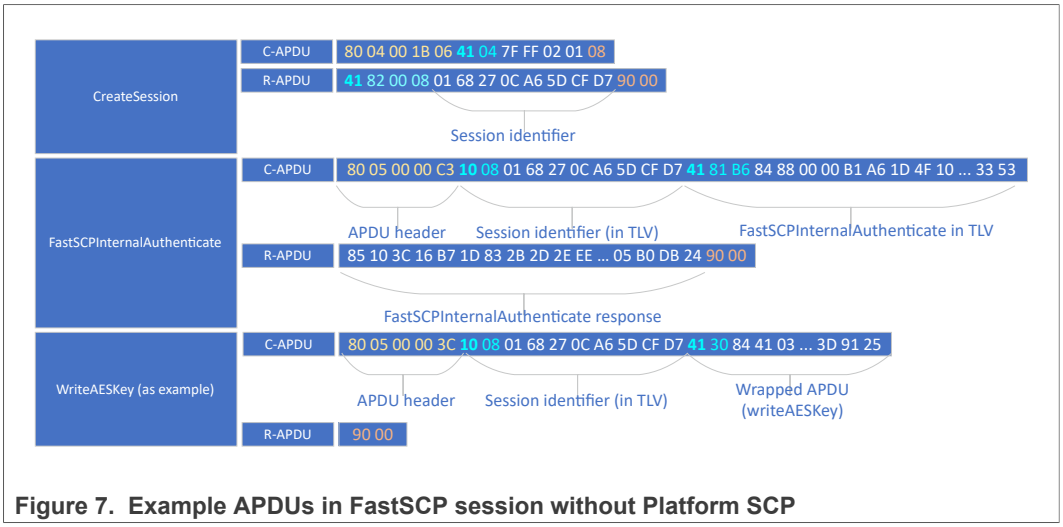

Figure 7. Example APDUs in FastSCP session without Platform SCP

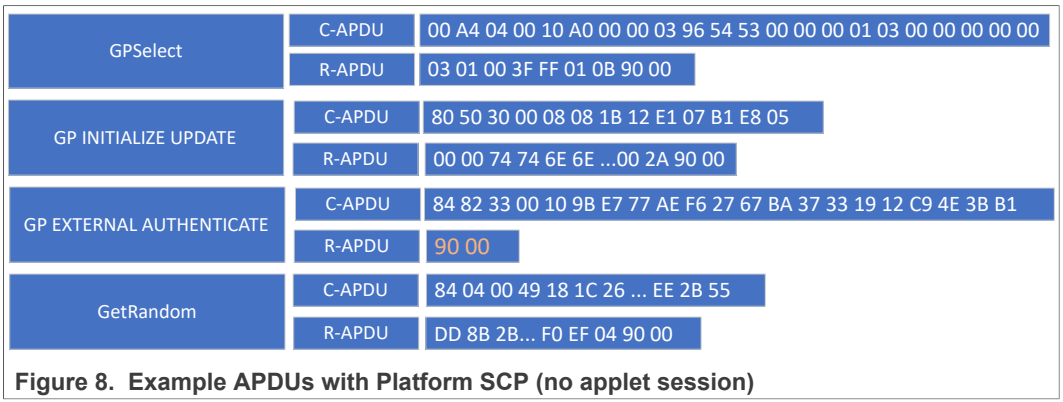

Figure 8. Example APDUs with Platform SCP (no applet session)

5.1.2.1 Session policies

Besides Secure Object policies, the user can also assign a policy to a session. If a policy is passed as argument during session creation, the session will be limited in its lifetime by the number of APDUs that are sent within the session. So the user passes the maximum number of APDUs and when this maximum is reached, the session is broken down and the user is no longer authenticated.

Note that the user can call SessionRefresh to extend the lifetime of a session.

5.1.2.2 Example UserID session

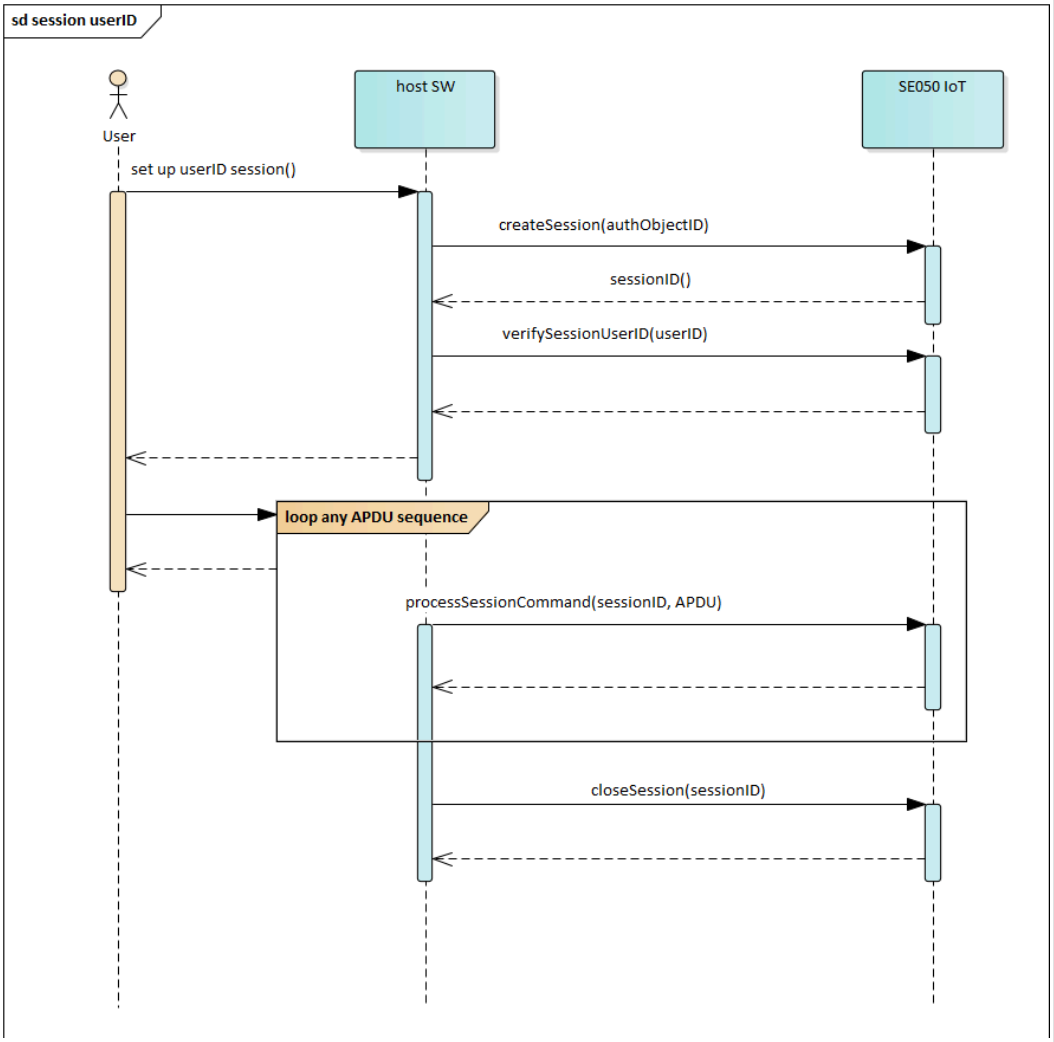

Figure 9. Example UserID session

Note: *Default session policy*

5.1.2.3 Example SCP03 session

Here the SCP03 protocol as defined in Global Platform (Card Specification v 2.2 – Amendment D) is used. The Global Platform SCP03 session is encapsulated within an applet session as shown in the picture below and is then called “applet SCP03 session”.

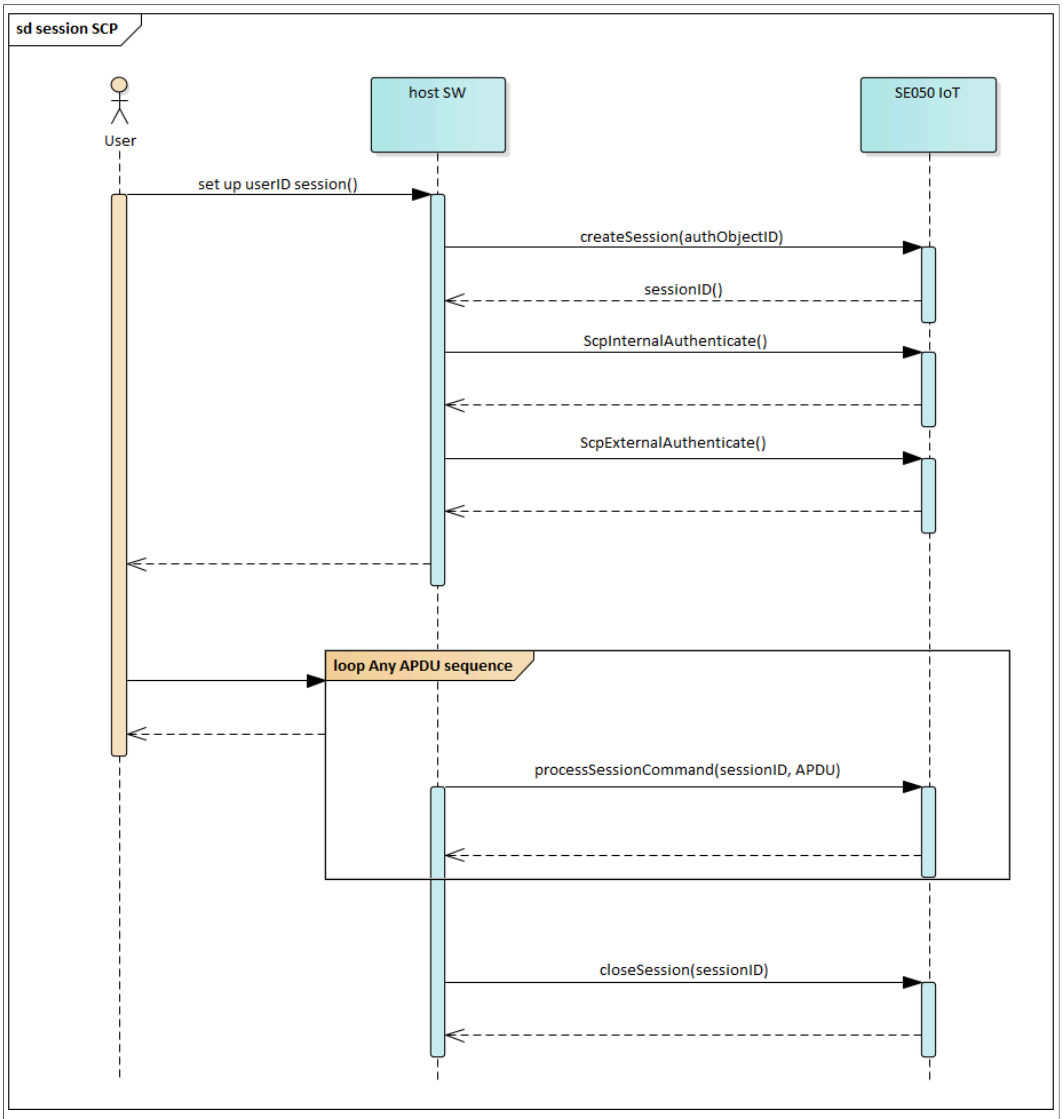

Figure 10. Example SCP session

Note: *Default session policy*

5.1.2.4 Example FastSCP session

FastSCP uses ECKeys stored in the SE050 to open an bound user session. The authentication is done using the asymmetric ECKey and the session is encrypted using the SCP03 protocol. The FastSCP authentication is defined in [1].

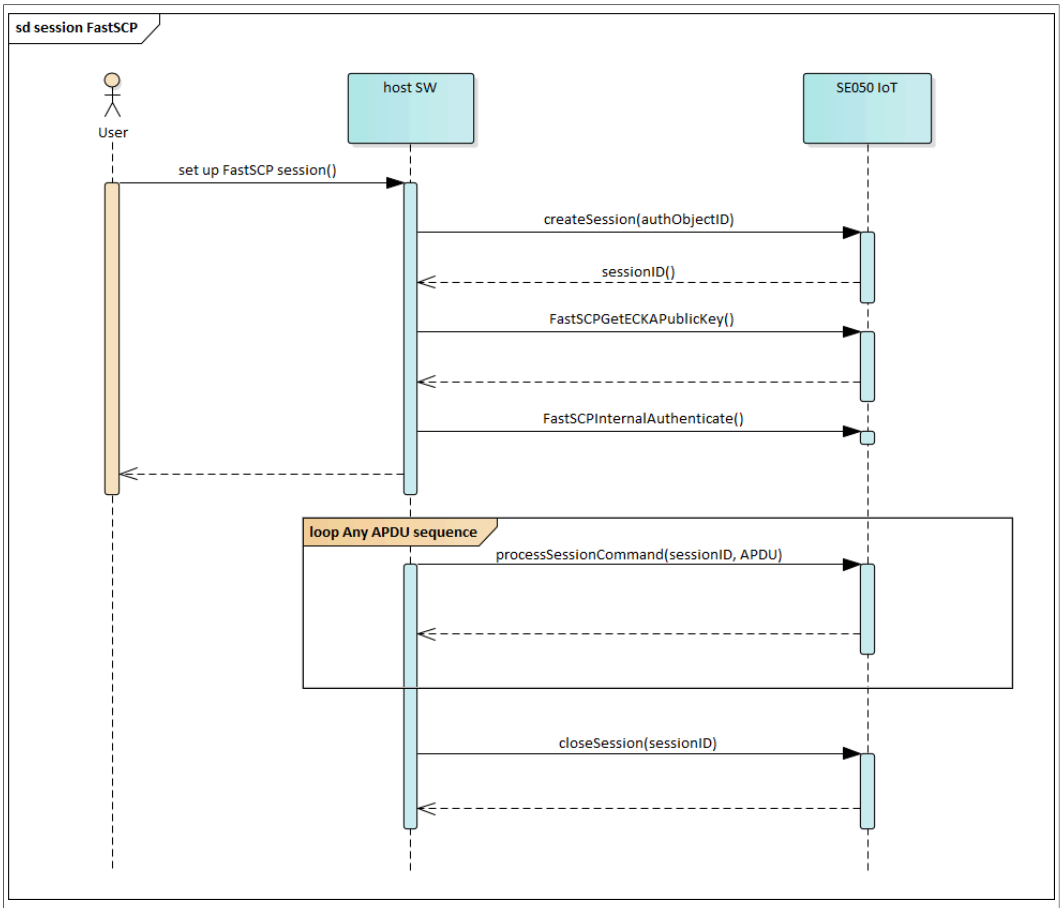

Figure 11. Example FastSCP session

Note: *Default session policy*

5.1.3 Secure Object policies for multi-tenant use

When multiple users are configured or expected, the Secure Object policies need to be set up to allow the right users to perform the right operations (and forbid to the wrong user).

As a (simple) example, there could be a mapping of users to object policies as follows:

Table 3. Secure Object policies for multi-tenant use

	User A	User B
POLICY_OBJ_ALLOW_DELETE	Granted	Denied
POLICY_OBJ_ALLOW_READ	Granted	Granted
POLICY_OBJ_ALLOW_WRITE	Granted	Denied
POLICY_OBJ_ALLOW_SIGN	Denied	Granted

6 Trust Provisioning

This chapter focuses on the credentials provisioning.

Credentials provisioning provides recommendations on how to provision credentials in SE050 in a secure manner.

6.1 Trusted or untrusted environments

The provisioning phase of the device is critical to the security of the product, since sensitive data and key material are being generated on or injected into the SE050.

A device can operate in different types of environments:

- **Trusted environment** – a trusted environment is a secured environment under control of a trusted party. The level of security depends on the environment and should be set appropriate to the threats and security objectives relevant to the device and its assets
- **Non-trusted environment** – a non-trusted environment is an unsecured environment and no control can be applied to it.

6.2 SE050 Trust Provisioning

The IoT device identity should be unique, verifiable and trustworthy so that device registration attempts and any data uploaded to the OEMs servers can be trusted. The SE050 is designed to provide a tamper-resistant platform to safely store keys and credentials needed for device authentication and registration to OEMs cloud service. Leveraging the SE050 security IC, OEMs can safely authenticate their devices without writing security code or exposing credentials or keys.

You can rely on any of the secure provisioning options for the SE050 security IC:

- **SE050 with Ease of Use configuration**: Every generic SE050 product variant comes with a specific set of credentials being trust-provisioned by NXP. These credentials can be used for all major use cases, including device-to-device authentication.
- **SE050 secure provisioning by NXP**: The NXP Trust Provisioning service offers customized and secure injection of die-individual keys and credentials into SE050 on behalf of the OEM. This service is available for high volume orders of more than 150K units.
- **SE050 secure provisioning by NXP distributors or third-party partners**: NXP has agreements with distributors and third-party partners to offer customized and secure injection of die-individual keys and credentials into SE050 for orders of any size.

Note: SE050 provisioning can optionally be done by the OEM in case it owns or invests in PKI infrastructure at their facilities.

7 Security Recommendations

This chapter describes requirements and recommendations which have to be followed to use the production in a secure manner. Not complying bears a risk of security gaps.

“Must”, “should” and “may” are used in compliance with RFC2119 (see [9].)

- **Must** indicates an absolute security requirement
- **Should** indicates a security recommendation, meaning there may exist valid reasons in particular circumstances to ignore a particular item

- **May** mean that an item is optional

7.1 Generic recommendations (all use cases)

7.1.1 Platform SCP

Either Platform SCP or SCP user session MUST be used to protect locally against eavesdropping.

Platform SCP keys MUST be updated at first use of the product.

Confidentiality, integrity and authenticity of the Platform SCP key set MUST be enforced as required during provisioning of the keys, outside of SE050, in the host and its memories.

- Platform SCP keys MUST be stored securely in the host.
- Access control on host keys MUST be enforced if available, or any mechanism that protects the keys from being divulgated.
- Further security properties MUST be preserved by environmental measures.

7.1.2 Initial State

Upon reception of SE050, the authenticity of the provisioned Secure Objects MAY be checked by reading the Trust Provisioned flag of the Secure Object.

- The presence of the Trust Provisioned flag ensures the secure object has been securely provisioned by NXP.

During SE050 lifecycle, the authenticity of the provisioned secure objects MAY be checked by reading the Secure Object content with attestation mode.

7.1.3 Attestation

When reading with attestation, the timestamp and freshness fields MUST be checked for each attestation to prevent reuse of attestation.

- The timestamp field for consecutive attestations MUST be checked to be consecutive.
- The freshness field for consecutive attestations MUST be checked to be different from each other.

If the user creates a key Secure Object for attestation, the Secure Object policy MUST NOT have the rule POLICY_OBJ_ALLOW_SIGN set.

7.2 Extendibility and Multi-tenant

7.2.1 Secure Object policy

If access control on Secure Object is required the user MUST set the policy of the object.

- Default policy grants all access to the Secure Objects. Binary files, counters, PCR are accessible in read and write mode. Cryptographic keys are accessible in write and generation mode.

7.2.2 Transport lock use

7.2.2.1 Transport lock

If a transport lock is expected in SE050 configuration, customer MUST verify that the lock is still applied upon receipt of SE050.

- In locked state, only GetVersion, GetUniqueID, GetRandom and CreateSession commands are allowed. Command ReadIDList should fail with response SW_COMMAND_NOT_ALLOWED.
- If an unexpected behavior is noticed, this MUST be reported to NXP.

If a transport lock is applied, SE050 MAY be unlocked. Unlocking response MUST be SW_NO_ERROR.

- Unlocking is achieved by authenticating to the reserved authentication object with identifier 0x7FFF0200. If this would fail or unexpected behavior is noticed, this MUST be reported to NXP.

7.2.2.2 Transport Lock Provisioning

These recommendations apply to customers or 3rd Party programming facilities who need transport lock to perform device provisioning.

The transport lock MAY be used as a tamper seal to distribute devices to other parties in a cascade logistic chain.

- In this scenario, the transport lock can be considered as seal, which hampers manipulations to the product during transport.

If more than one customer is intended to perform provisioning in the supply chain, each customer MUST update the transport lock.

- In this case the transport lock MUST be configured with write access policy.

7.2.3 UserID sessions

UserID sessions are not providing authentication functionality but are there to logically group Secure Objects.

7.2.3.1 UserID Secure Object

For secure use of UserID Secure Objects, the maximum authentication attempts TAG_MAX_ATTEMPTS MUST be set to a value different from zero.

- A UserID Secure Object TAG_MAX_ATTEMPTS with a value of zero means infinite authentication attempts. The UserID can be compromised by brute-force attack.
- Note that a TAG_MAX_ATTEMPTS with a value different from zero will cause a flash write at each UserID verification due to counter pre-decrement.

7.2.3.2 Security claims on user sessions

UserID sessions MUST NOT be used alone if confidentiality or integrity of communications are required.

- UserID sessions are not secure intrinsically as the UserID can be eavesdropped, and following communications are not encrypted.

- To ensure confidentiality and integrity of communications SCP03 or FastSCP sessions MUST be used.

7.2.3.3 Secure messaging

If confidentiality is required on a secure object, the Secure Object policy MUST either have the rule POLICY_OBJ_REQUIRE_SM set, or have the Authentication Object ID referring to an existing key authentication object (AES128 key for an SCP03 session or ECKey for a FastSCP session) or Platform SCP has to be configured mandatorily.

- These policies enforce use of secure messaging and thus ensure confidentiality and integrity of communications.

7.2.4 Credentials provisioning

These recommendations apply to customers or 3rd Party programming facilities who perform provisioning

Note: Provisioning can be protected/restricted by establishing authenticated session with restricted access rights.

7.2.4.1 Remote provisioning

For transferring secret key during remote provisioning, applet level SCP or Secure Object Import MUST be used.

- Use of applet level SCP (SCP03 or FastSCP sessions) or Secure Object Import in addition to Platform SCP is mandatory to ensure end-to-end confidentiality and integrity of secrets during remote provisioning.

7.2.4.2 Key pairs

Confidentiality, integrity and authenticity of key pairs that are provisioned into SE050 MUST be enforced as required for their use during provisioning and outside of SE050.

- Key pairs MAY be generated on-chip so that private keys can stay in SE050.
- Further security properties MUST be preserved by environmental measures.

If only integrity and authenticity of key pairs that are provisioned into SE050 is required, attestation of the Secure Object with a provisioned attestation key MAY be used.

Key pairs provisioned into SE050 MUST be die-individual.

- Key pairs fights exploitation of successful attacks on other devices.

7.2.4.3 Symmetric keys

Confidentiality, integrity and authenticity of symmetric secrets that are provisioned into SE050 MUST be enforced as required for their use also during provisioning and outside SE050.

- SE050 supports Symmetric Keys with self-wrapped input of a symmetric secret according to RFC3394. Note that keys MUST be protected as required before the wrapping.
- Further security properties MUST be preserved by environmental measures.

If only integrity of symmetric keys that are provisioned into SE050 is required the attestation of the Secure Object with a provisioned attestation key MAY be used.

- The timestamp and freshness fields of the attestation must be checked according to Attestation.

When the use case allows it, symmetric secrets provisioned into SE050 SHOULD be die-individual.

- Symmetric Keys fights exploitation of successful attacks on other devices.

7.2.5 General purpose storage

Integrity and authenticity of GP data that are provisioned into SE050 MUST be enforced as required for their use during provisioning and outside of SE050.

- Integrity of GP data that are provisioned into SE050 is supported by attestation of the GP data with a provisioned attestation key.
- Further security properties MUST be preserved by environmental measures.

8 Functional Recommendations

8.1 Wear-out prevention

NVM writes have the risk to wear out the flash and thus permanently make the device unusable).

The default configuration of the secure element avoids as much as possible NVM writes: only when storing keys or files permanently into the device, flash write operations are done.

Creation and deletion of any Secure Object or Crypto Object is causing flash write operations. For transient Secure Objects and for Crypto Objects, any update of the value of the Secure Object is not causing additional flash write operations. For persistent Secure Objects, any update of the value of the Secure Object causes flash write operations.

Additional flash writes are done when users opt for putting a maximum number of authentication attempts on Authentication Objects. In that case, any authentication attempt is logged and causing additional flash write operations.

An exception to the general rules above is the shared secret generation in case the EC Montgomery curve 25519 is used: the externally provided public key will in that case be stored in NVM as well, so each shared secret generation will cause additional NVM write operations as well to store the external public key that is used in the shared secret generation.

8.1.1 Power modes of SE050

SE050 supports the following kinds of power saving operations:

- “Off”: For this scenario, Vin is not supplied anymore. As a consequence, the IC looses all its internal states that are not yet persisted in NV memory. A full startup sequence needs to be executed.

- “Deep Power Down” via ENA pin: This mode is entered if the ENA pin is de-asserted. The behavior from IC point of view is identical to the power down mode described above. All transient states are lost.
- “Power Down”: When using I^2C interface, a “end of command apdu” command can be sent on the I^2C link to bring the device into a sleep mode. In that mode, all transient states are kept and the communication can continue with the next APDU
- Active mode: This mode is automatically entered when waiting for next command apdu.

Depending on the startup performance and power saving requirements one of the modes above should be chosen when the SE050 is not actively used.

8.1.1.1 Application Circuit Basic I2C usage

Configuration used:

- I^2C from host to SE050
- SE050 turned off via V_{cc} off
- Contactless and I^2C master not part of application schematic below
- Alternatively V_{cc} can be connected to V_{in} instead of V_{out}. V_{out} is then not connected in this case.

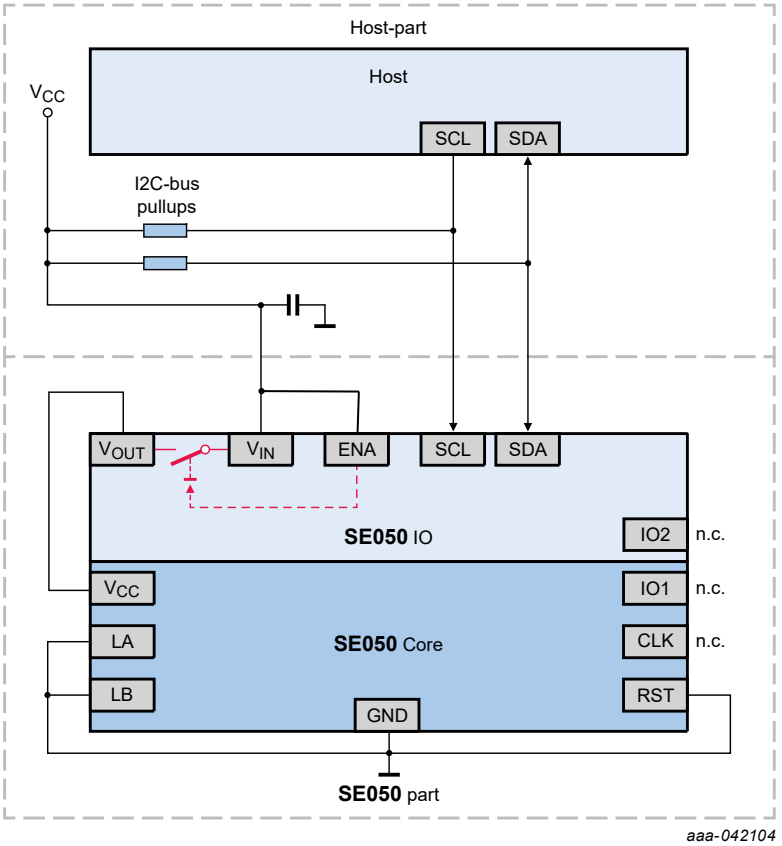

Figure 12. Application Circuit I^2C

8.1.1.2 Application Circuit I^2C with Deep Power Down

Configuration used:

- I^2C from host to SE050

- SE050 turned off using Deep Power Down mode (ENA pin low)
- Contactless and I²C master not part of application schematic below

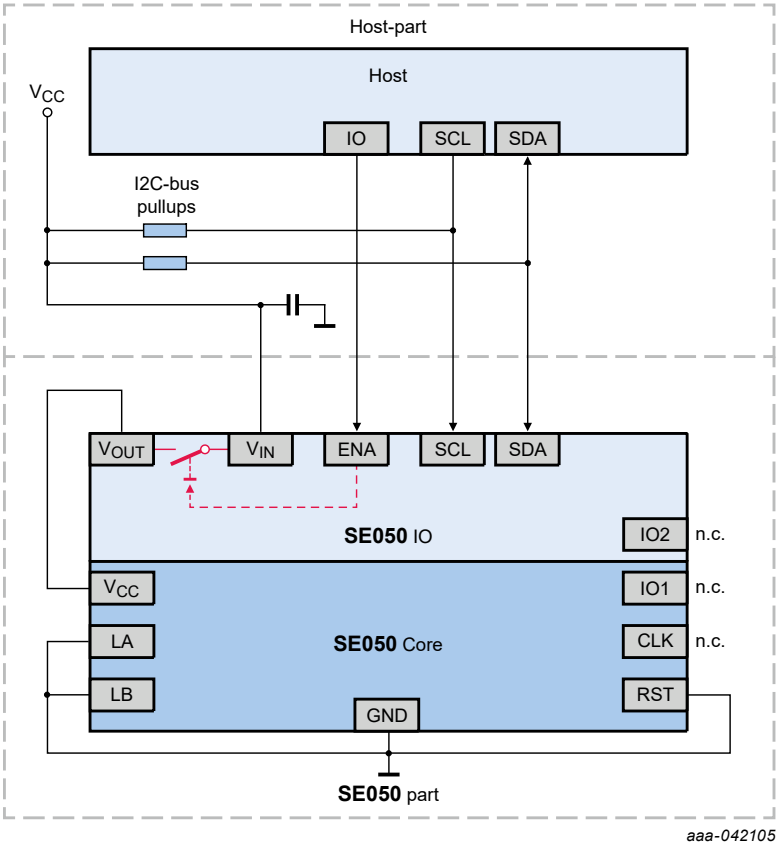

Figure 13. Application circuit I2C with Deep Power Down

No extra capacitor to be placed on the connection between V_{out} and V_{cc}.

8.1.1.3 Application Circuit I²C Master (with Deep Power Down)

Configuration used:

- I²C from host to SE050
- SE050 turned off using Deep Power Down mode (ENA pin low)
- I²C Master interface connected, in the below application schematic external sensor supplied by V_{out} of SE050
- Contactless and I²C master not part of application schematic below

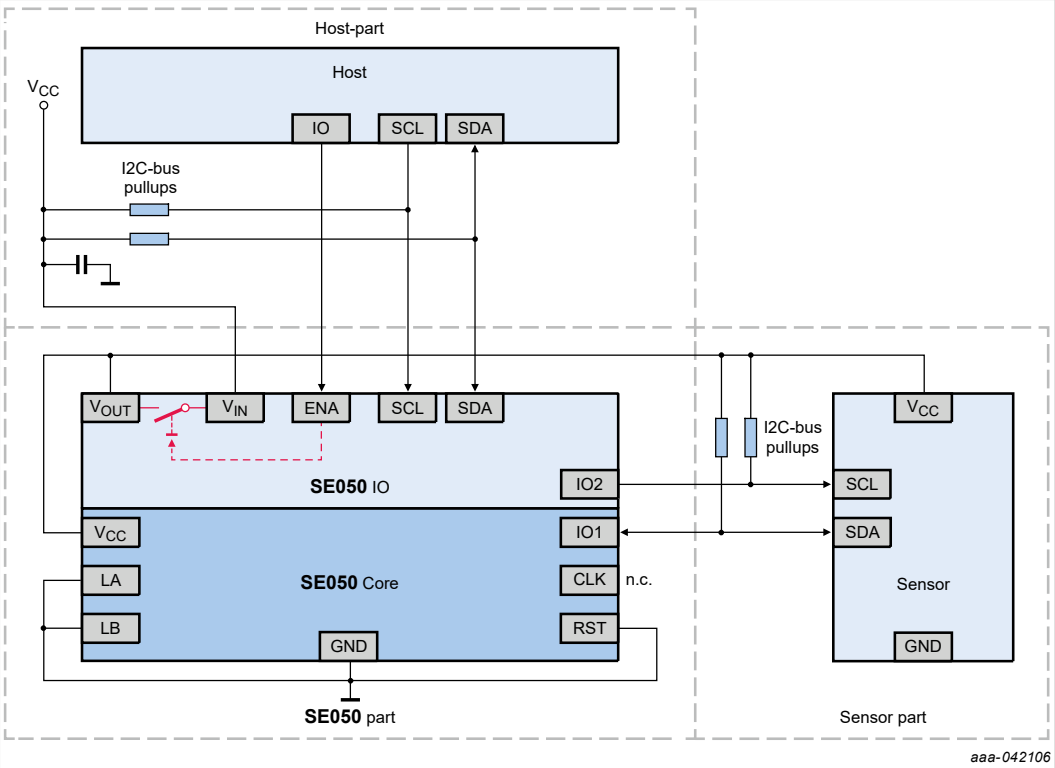

Figure 14. Reference Schematic for Smart Sensor

8.1.1.4 Application Circuit Contactless only

The resonance frequency of the antenna in the assembled system should maintain a range between 13.6 to 14.1 MHz in a field strength in which the device operates. The target system resonance frequency is 13.8 MHz.

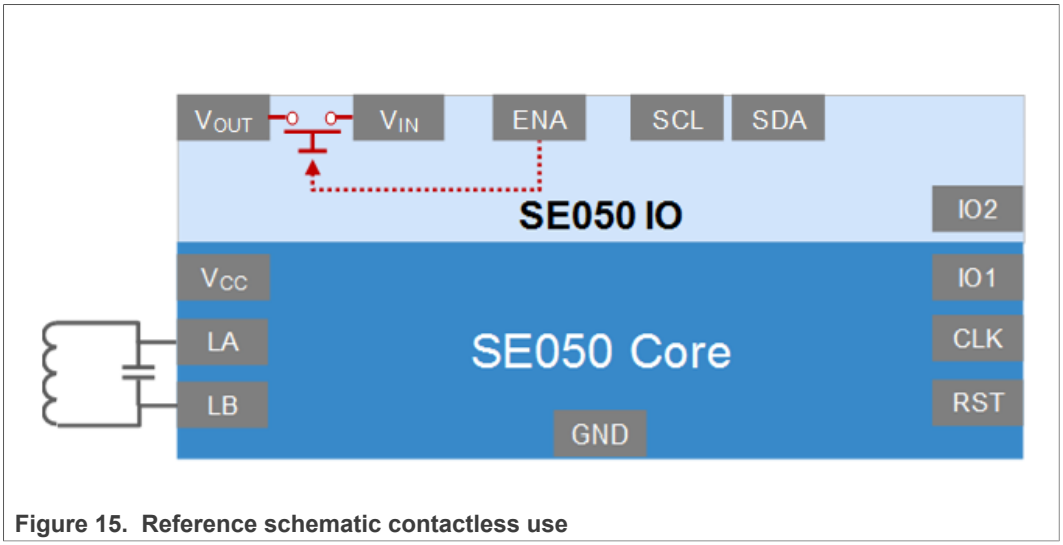

Figure 15. Reference schematic contactless use

Due to detuning effects, avoid ferromagnetic metal or metal foils in the range of the antenna. If you need to guide the HF magnetic flux in the housing, a reduction of the count of windings is necessary and a capacitor in parallel for tuning might be necessary.

Ensure to minimize the length from the terminals of the antenna to the IC and route the traces next to each other to the SE050.

Remove GND plane under the antenna and maintain >5mm distance from other traces and GND plane. The GND plane reduce the performance and detune the antenna. Avoid material with high permittivity in proximity of the windings of the coil, e.g. housing. (maintain larger >2mm) The additional permittivity generates an additional capacitance which influences the resonance frequency.

It is recommended to add footprints for spare passive components in parallel to the inductor , C) for optimization of read ranges in combination with mobile phones. The rating of this components is expected to be <10V, current < 40 mA.

9 References

1. Application Note: SE050 IoT Applet APDU Specification, AN 12413
2. Application Note: SE050 Configurations, AN12436. Available under: https://www.nxp.com/docs/en/application-note/AN12436.pdf
3. Application Note: SE050 Secure Connection Azure IoT Hub, AN12402. Available under: https://www.nxp.com/docs/en/application-note/AN12402-SE050_secure_connectiom_azure_iot_hub.pdf
4. Application Note: SE050 Secure Connection AWS IoT Core, AN12404
5. Application Note: SE050 Secure Connection OEM Cloud, AN12400. Available under https://www.nxp.com/docs/en/application-note/AN12400-SE050_secure_connection_OEM_cloud.pdf
6. Application Note: SE050 Secure Connection Google Cloud IoT Core, AN12401. Available under: https://www.nxp.com/docs/en/application-note/AN12401-SE050_secure_connection_google_cloud_iot_core.pdf
7. Application Note: SE050 Secure Connection Watson IoT Device to Device Authentication, AN12403. Available under: https://www.nxp.com/docs/en/application-note/AN12403.pdf
8. Application Note: Device to Device authentication, AN12399. Available under: https://www.nxp.com/docs/en/application-note/AN12399-SE050_device_to_device_authentication.pdf
9. RFC2119. Available under: https://tools.ietf.org/html/rfc2119

10 Legal information

10.1 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.

10.2 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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

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 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.

10.3 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.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 November 2021
Document identifier: AN12514