AN12660

Ease ISA/IEC 62443 compliance with EdgeLock^{TM} SE050

Rev. 1.0 — 16 June 2020 Application note
582810

Document information

Information	Content
Keywords	ISA/IEC 62443, Industrial security, EdgeLock SE050
Abstract	This document elaborates on the use of EdgeLock SE050 features to reduce implementation complexity and to fulfill the security requirements mandated by the ISA/IEC 62443-4-2 standard.

Revision history

Revision history

Revision number	Date	Description
1.0	2020-06-16	Initial version

1 Introduction

The potential risk from cyberattacks increases as the number of connected controllers, machines, devices and sensors keeps growing. As such, security proves itself as a critical element in the development of industrial control systems against intentional or unintentional threats. These threats may include personal injury, equipment damage, supply chain downtime, environmental impact, loss of production or violation of regulatory requirements, among others.

The industry has responded to cybersecurity threats by creating standards to assist end-users and equipment vendors through the process of securing industrial control systems. In this respect, the ISA/IEC 62443 series of standards addresses the security of Industrial Automation and Control Systems (IACS) throughout their lifecycle.

With ISA/IEC 62443 certification, OEMs demonstrate that their systems or products have been independently evaluated to ensure that they are free from known vulnerabilities and have a robust architecture for protection against cyber attacks. In addition, it provides assurance and confidence to end-users that products comply with higher standards for employee safety.

As part of the ISA/IEC 62443 standard, four security levels (SL1, SL2, SL3 and SL4) are defined, each of which represents an incremental level in terms of cybersecurity measures and in the requirements to be met. In this context, the use of a Secure Element (SE) such as EdgeLock SE050 with its pre-integrated security features eases the compliance with ISA/IEC 62443 component requirements and it allows the OEM to strengthen even more the IoT device against logical and physical attacks, making the device future-proof.

2 How to use this document

This document is addressed to OEMs interested in understanding how EdgeLock SE050 can be used to facilitate the implementation of ISA/IEC 62443-4-2 requirements. It is structured as follows:

- ISA/IEC 62443 standard overview section provides a brief introduction to ISA/IEC 62443 standard and its main concepts.
- Leverage EdgeLock SE050 to meet ISA/IEC 62443-4-2 requirements section elaborates on a set of security primitives needed in order to achieve ISA/IEC 62443-4-2 compliance, and describes how EdgeLock SE050 can be leveraged to meet ISA/IEC 62443-4-2 requirements.
- ISA/IEC 62443-4-2 requirements lookup table section maps ISA/IEC 62443-4-2 requirements with the associated security primitives helping to meet that particular requirement.
- The Glossary section lists common acronyms used throughout the document and defines their meaning.

3 ISA/IEC 62443 standard overview

The ISA/IEC 62443 standard is a series of standards and technical reports helping organizations to mitigate the risk of failure and exposure to security vulnerabilities in Industrial Automation and Control Systems (IACS). The ISA/IEC 62443 is organized into four categories: *General*, *Policies and Procedures*, *System*, and *Component*:

- *General information*: its four parts contain foundational information such as concepts, models and terminology used as a basis for the other categories of the standard.
- *Policies and Procedures*: its five parts comprise the requirements and different aspects for creating and maintaining effective security processes. This part of the standard specifically targets factory operations.
- *System*: its three parts describe the technical requirements for system design and the guiding principles for implementing and integrating secure systems. This part of the standard is targeted to industrial system integrators.
- *Component*: its two parts contain the technical guidelines for developing secure industrial components or products. This part of the standard is targeted to manufacturers of industrial devices.

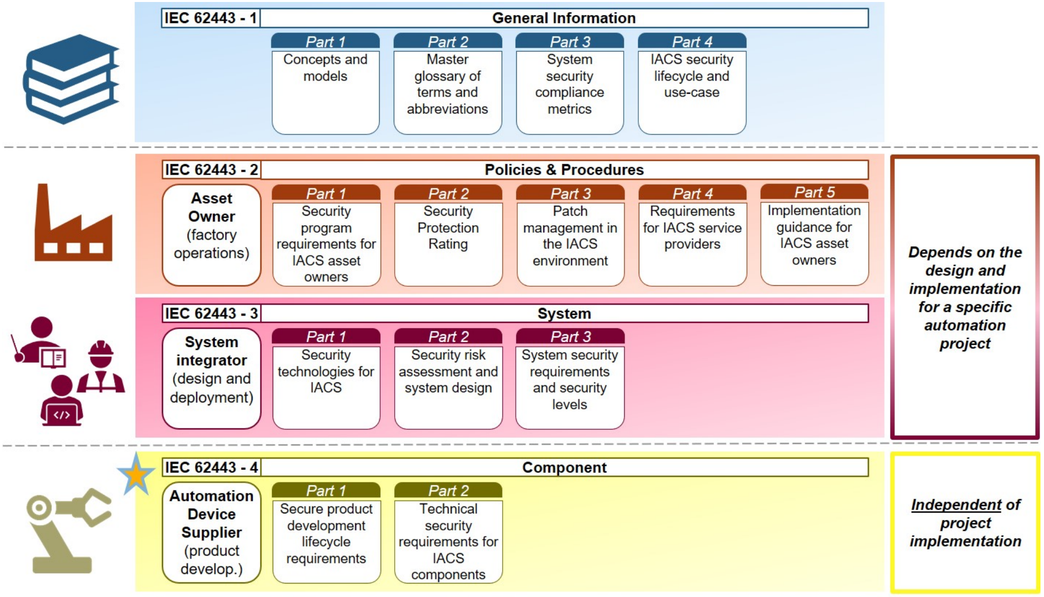
IEC 62443 - 1
General Information
Part 1
Concepts and models
Part 2
Master glossary of terms and abbreviations
Part 3
System security compliance metrics
Part 4
IACS security lifecycle and use-case
IEC 62443 - 2
Policies & Procedures
Asset Owner (factory operations)
Part 1
Security program requirements for IACS asset owners
Part 2
Security Protection Rating
Part 3
Patch management in the IACS environment
Part 4
Requirements for IACS service providers
Part 5
Implementation guidance for IACS asset owners
IEC 62443 - 3
System
System integrator (design and deployment)
Part 1
Security technologies for IACS
Part 2
Security risk assessment and system design
Part 3
System security requirements and security levels
Depends on the design and implementation for a specific automation project
IEC 62443 - 4
Component
Automation Device Supplier (product develop.)
Part 1
Secure product development lifecycle requirements
Part 2
Technical security requirements for IACS components
Independent of project implementation

Table taken from ISA/IEC 62443 *IACS: Industrial Automation Control system (analog expression for ICS)

Figure 1. ISA/IEC 62443 overview

To assess and classify the required protection level, the ISA/IEC 62443 standard defines the concept of security assurance levels. These security levels are connected to risk and asset value and are organized in tiers, each one requiring more stringent measures to be put in place, as detailed in Table 1:

Table 1. ISA/IEC 62443 security assurance levels

Security level (SL)	Description from ISO/IEC 62443-1-1 section 10.4.3
SL0	No specific requirements or security protection requirements
SL1	Requires protection against casual or coincidental violations
SL2	Requires protection against intentional violation using simple means with low resources, generic skills and low motivation
SL3	Requires protection against intentional violation using sophisticated means with moderate resources, specific skills and moderate motivation
SL4	Requires protection against intentional violation using sophisticated means with extended resources, specific skills and high motivation

Security Levels 1 and 2 correspond to threats originating from either insiders or intruders with low skills and motivation. On the other hand, Security Levels 3 and 4 are related to threats from “professional” cyber criminals, industrial espionage or state-sponsored malicious actors that demonstrate high skills and moderate to high motivation.

The ISA/IEC 62443 standard establishes a practical guide on how to implement protective measures against cybersecurity incidents based on the defined security levels, grouped into seven foundational requirements:

- FR1: Identification and Authentication Control (IAC)
- FR2: Use Control (UC)
- FR3: System Integrity (SI)
- FR4: Data Confidentiality (DC)
- FR5: Restricted Data Flow (RDF)
- FR6: Timely Response to Events (TRE)
- FR7: Resource Availability (RA)

Each foundational requirement (FR) defines specific security requirements depending on component type, scope and applicability. The requirements that apply indifferently to all component types are denoted as Component Requirements (CR). In case a requirement applies only to a specific component type, the requirement is denoted as Embedded Device Requirement (EDR), Network Device Requirement (NDR), Software Application Requirement (SAR) or Host Device Requirement (HDR) accordingly.

Table 2 details the component types defined in ISA/IEC 62443 standard.

Table 2. Component types

Type	Description	Example
Embedded Device (ED)	Specialized device designed to directly monitor, control or actuate an industrial process. An embedded device typically runs embedded software, has an embedded OS or firmware and can be programmed only through external interfaces. It may have a communication interface and an attached control panel.	Programmable Logic Controller (PLC), Safety Instrumented System (SIS) controller, Distributed Control System controller (DCS)
Network Device (ND)	Device that facilitates or restricts the data flow between devices, but does not directly interact with a control process.	Firewall, router, gateway, switch
Host Device (HD)	General purpose device running a general purpose OS (for example Microsoft Windows OS or Linux). A host device is capable of running one or more general purpose applications and it typically has a local or remote human-machine interface. **Note:** Host devices are outside the scope of this document.	Server, PC, data centers
Software Application (SA)	Software program that is used to interface with the process or the control system. It is typically executed in embedded devices and host devices.	SCADA software, PLC ladder-programming software, data loggers

As a conclusion, the ISA/IEC 62443 standard provides a point of reference for all the actors participating in the IACS ecosystem to improve cybersecurity in industrial environments. On this basis, OEMs and manufacturers can implement the protective measures to comply with the necessary requirements to achieve the target security level.

4 Leverage EdgeLock SE050 to meet ISA/IEC 62443-4-2 requirements

The various ISA/IEC 62443 standards depicted in Figure 1 are intended to be multi-industry in nature and are targeted at different audiences, ranging from suppliers and device vendors to end-users. For OEMs of industrial products, including applications, embedded devices, network components and host systems, the security functions required at the component level are listed in ISA/IEC 62443-4-2.

Figure 2 depicts a simplified breakdown of an industrial control system integrating a smart, connected industrial device, also referred to as IoT device, that leverages EdgeLock SE050. The IoT device is represented by the *platform*, composed of an MCU / MPU connected to EdgeLock SE050, and the *application*, which is the software running in the IoT device hardware. This IoT device is integrated within an industrial control system, which includes its own network resources, backend servers and operational processes. The different parts of the IoT device are typically strictly integrated and their combined features allow a component to achieve the security requirements imposed by ISA/IEC 62443-4-2.

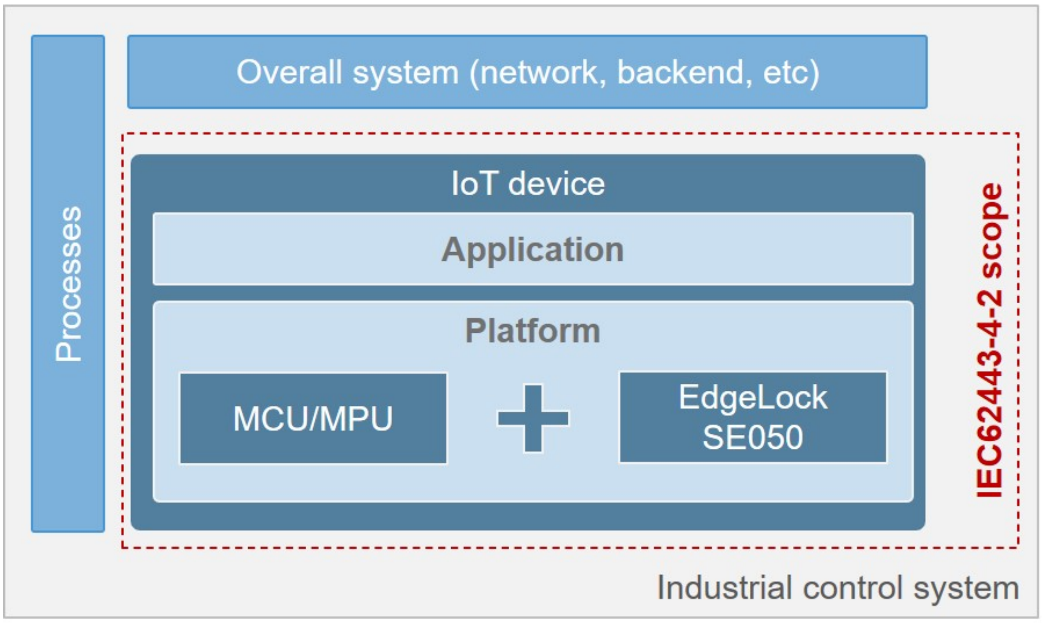

Figure 2. Breakdown of a simplified industrial control system

Note: *In the scope of ISA/IEC 62443-4-2 standard, the focus is on the component level and the security features to be implemented at the IoT device level. Further elements in industrial control systems are not considered.*

EdgeLock SE050 offers a trusted, highly secure environment where critical keys and credentials can be stored securely and where built-in cryptographic operations using secure cryptographic algorithms can be performed.

EdgeLock SE050 simplifies the implementation of security features in industrial system components since it allows to outsource to a single chip many of those security-related operations that would otherwise require a complex software implementation. In this respect, EdgeLock SE050 comes with a pre-installed IoT applet offering advanced key management and cryptographic functions. To ease the integration of the applet functionalities in the IoT solution, EdgeLock SE050 even provides a fully-featured middleware package. The middleware is pre-integrated with many micro-controller platforms and contains several examples and demo projects that can be used as a starting point for custom software implementations.

Moreover, EdgeLock SE050 is pre-provisioned with keys and credentials in a highly secure and controlled environment, therefore relieving IoT device manufacturers from setting up a complex and expensive PKI infrastructure.

As a result, EdgeLock SE050, in combination with MCUs / MPUs and software applications, acts as an enabler of the security requirements defined in ISA/IEC 62443-4-2 and even more, it goes beyond what is strictly required by the standard, and provides an extra level of security that makes an IoT device future-proof and resistant to the latest security threats.

In order to present a more organic view of security requirements, we identified a set of security primitives (SP) with the purpose of defining a common and easier to understand nomenclature across standards to describe security requirements in IoT systems.
In respect of ISA/IEC 62443 standard, security primitives help you to map security features of an IoT device to ISA/IEC 62443-4-2 security requirements. You can find more information about how to use security primitives in the white paper Security Primitives: Common Nomenclature to Describe Security Requirements in IoT Systems.

The security primitives where EdgeLock SE050 can add a valuable contribution are listed in Table 3. In the next sections, EdgeLock SE050 features will be put in the context of each security primitive listed in Table 3. Which ISA/IEC 62443-4-2 requirements EdgeLock SE050 can help to achieve will also be described.

Table 3. Security primitives definition

Code	Security primitive
SP1	Anomaly detection and reaction
SP2	Device attestation
SP3	Secure backup and recovery
SP4	Protection of Personal Information
SP5	Secure Provisioning and Decommissioning
SP6	Cryptographic Random Number Generation
SP7	Root of Trust
SP8	Secure Communication Protocols
SP9	Secure Initialization
SP10	System Event Logging
SP11	Secure Encrypted Storage
SP12	Cryptographic Key Generation and Injection
SP13	Cryptographic Key and Certificate Store
SP14	Cryptographic Operation

Code	Security primitive
SP15	Secure Onboarding and Offboarding
SP16	Secure Updates

4.1 SP1: Anomaly detection and reaction

The security primitive *Anomaly Detection and Reaction* clusters software and hardware features that monitor the IoT device for abnormal events and, if required, trigger and execute an appropriate action. Typically, these actions encompass logging the anomaly, issuing a message to the cloud backend, resetting the device, and/or changing a secure lifecycle state. This primitive includes logical and physical tamper detection and tamper protection of the IoT device.

EdgeLock SE050 provides enhanced run-time integrity protection for IoT devices. By pairing it with the host MCU of the device, a secure initialization of the device can also be enforced (see Binding an MCU to EdgeLock SE050 Application Note). Any anomaly occurring during that stage would result in a lifecycle state of the Secure Element (SE) in which mission critical keys and certificates are not available to the host device. Further details on how the EdgeLock SE050 ensures that long-lived credentials are kept safe during the device lifecycle are provided in Cryptographic Key and Certificate Store security primitive. In addition, EdgeLock SE050 provides a tamper-resistant platform, certified up to the Operating System (OS) level at CC EAL 6+. In case a physical tampering of the SE is detected, EdgeLock SE050, in combination with specific software application components, can trigger the appropriate countermeasures, such as notifying the system backend or resetting the device.

Leveraging both mechanisms of ensuring run-time integrity and triggering actions on detected anomalies provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 4 to the highest security levels.

Table 4. ISA/IEC 62443-4-2 requirements supported by SP1 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.5.1	Hardware security for authenticators	-	-	X	X
CR 1.9.1	Hardware security for public key-based authentication	-	-	X	X
CR 1.14.1	Hardware security for symmetric key-based authentication	-	-	X	X
CR 3.4.2	Automated notification of integrity violations	-	-	X	X
EDR/NDR 3.11.0	Physical tamper resistance and detection	-	X	X	X
EDR/NDR 3.11.1	Notification of a tampering attempt	-	-	X	X

EdgeLock SE050 inherently supports the ISA/IEC 62443 requirement EDR/NDR 3.11.0 with its integrated tamper protections certified up to the OS level at CC EAL 6+ including AVA_VAN 5, the highest achievable level in vulnerability analysis and penetration testing. Additionally, application developers can leverage EdgeLock SE050 tamper reaction features to easily fulfil CR 3.4.2 and 3.11.1. Finally, if authenticator keys are stored inside the EdgeLock SE050 key store, the requirements CR 1.5.1, CR 1.9.1 and CR 1.14.1 are inherently fulfilled (and certified). In this context, EdgeLock SE050 ensures that long-lived credentials are kept safe during the device lifecycle, even if an attacker has physical access to the device. Cryptographic operations are always performed inside EdgeLock SE050 with the keys remaining in the secure environment.

4.2 SP2: Device attestation

The *Device attestation* security primitive clusters those features that provide evidence of the IoT device genuine identity, its software and firmware version, as well as its integrity and lifecycle state. Genuine identification requires ensuring a unique identification of the IoT device.

EdgeLock SE050 is pre-injected in NXP's secure facilities with a device-unique, read-only 7-byte UID that can be used to identify the whole IoT device. If the use case requires it, a custom identifier can also be injected in EdgeLock SE050 and protected against deletion and overwriting using the appropriate policies. EdgeLock SE050 also supports storage of X.509 certificates that can be used to bind the device identity to a public key. Such certificates can be used to attest the device identity as part of challenge-based protocols that assess the possession of the corresponding private key. Certificates can be stored in DER format as binaries and protected against deletion and overwriting using the appropriate policies. EdgeLock SE050 is pre-provisioned with a set of keys, and corresponding certificates, that can be used to attest the device identity in different use cases, including cloud onboarding and device-to-device authentication. More information can be found in Secure Onboarding and Offboarding security primitive.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the industrial IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Read EdgeLock SE050 pre-injected UID:** Se05x_API_ReadObject (kSE05x_AppletResID_UNIQUE_ID)
- **Read binary identifier / certificate**: sss_key_store_get_key (), Se05x_API_ReadObject ()
- **Inject a binary identifier / certificate**: sss_key_store_set_key (), Se05x_API_WriteBinary ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to show how to use the pre-installed applet and ease the implementation of the use cases supported by the *Device attestation* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **EdgeLock SE050 Get Info example:** \simw-top\demos\se05x\se05x_Get_Info (see Section 5.15 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Get certificate from the SE example:** \simw-top\demos\se05x\se05x_Get_Certificate (see Section 5.18 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Building a self-signed certificate demo:** \simw-top\demos\se05x\certificate_demo (see Section 5.37 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging identifier and certificate management capabilities provided by EdgeLock SE050 aids in achieving ISA/IEC 62443-4-2 compliance for the requirements listed in Table 5 at the highest security levels.

Table 5. Requirements eased by SP2 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.2.0	Software process and device identification and authentication	-	X	X	X
CR 1.2.1	Unique identification and authentication	-	-	X	X

EdgeLock SE050 supports CR 1.2.0 and CR 1.2.1 by providing a pre-injected 7-byte UID identifier and a set of digital certificates uniquely bound to the device. Identifiers and certificates can be used to attest the genuine identity of the IoT device.

4.3 SP3: Secure backup and recovery

The *secure backup and recovery* security primitive clusters those functionalities that are used to back up the device (locally or in the cloud), and/or restore it at a later point in time. The backup may include user data, device software, device state, device configuration, or a combination thereof. The backup data shall be integrity and authenticity protected.

EdgeLock SE050 provides support for cryptographic signature algorithms that can be used to sign and then verify the digest of a backup in order to ensure the backup integrity and authenticity before restoring it. EdgeLock SE050 supports both RSA and ECC (ECDSA, EdDSA) signature algorithms for this purpose. The backup digests can be generated by EdgeLock SE050 using supported hash functions (SHA-224 to SHA-512). For additional protection of the backup data, EdgeLock SE050 can be leveraged to encrypt the backup before saving it locally or uploading it to the cloud. EdgeLock SE050 provides both asymmetric encryption algorithms (RSA, ECC) and symmetric encryption algorithms (AES, DES) for this purpose. The EdgeLock SE050 tamper-resistant hardware ensures that signing keys and encryption keys are protected.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Sign / verify a digest of a backup:** sss_se05x_asymmetric_sign_digest (), sss_se05x_asymmetric_verify_digest (), sss_se05x_asymmetric_sign (), sss_se05x_asymmetric_verify ()
- **Generate a digest of a backup:** sss_se05x_digest_one_go ()
- **Encrypt / decrypt a digest of a backup:** sss_se05x_asymmetric_encrypt(), sss_se05x_asymmetric_decrypt(), sss_se05x_cipher_one_go()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Secure Backup and Recovery* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Message Digest Example:** \simw-top\sss\ex\md (see Section 5.2.6 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECC Signing Example:** \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA Signing Example:** \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging encryption and signature functions provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 6 at the highest security levels.

Table 6. ISA/IEC 62443-4-2 requirements supported by SP3 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 7.3.1	Backup integrity verification	-	X	X	X

EdgeLock SE050 supports CR 7.3.1 by providing cryptographic functions to generate a digest of the backup, signing it and verifying the signature before the backup is restored. Additionally, EdgeLock SE050 supports encryption and decryption of the backup content in order to ensure the confidentiality of sensitive backup data. Cryptographic keys are always securely stored in EdgeLock SE050 secure tamper-resistant hardware and never leave the boundaries of the SE.

4.4 SP4: Protection of personal information

The security primitive *Protection of Personal Information* clusters those features that help preserve the confidentiality of personally identifiable information of end users that might be stored or managed by the IoT device.

EdgeLock SE050 provides support for cryptographic algorithms that can be used to encrypt sensitive information and protect it from unauthorized access and disclosure. EdgeLock SE050 supports both symmetric (DES, AES) and asymmetric encryption (RSA, ECC) for this purpose. EdgeLock SE050 implements strong hardware security for protecting encryption keys. In combination with proper user authorization implemented at the application level, EdgeLock SE050 can be used to grant access to sensitive information only to authorized users.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Encrypt sensitive personal information:** sss_se05x_asymmetric_encrypt (), sss_se05x_cipher_one_go ()
- **Decrypt sensitive personal information:** sss_se05x_asymmetric_decrypt (), sss_se05x_cipher_one_go ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Protection of Personal information* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)

EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 7 at the highest security levels:

Table 7. ISA/IEC 62443-4-2 requirements supported by SP4 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.5.1	Hardware security for authenticators	-	-	X	X

Code	Requirement	SL1	SL2	SL3	SL4
CR 4.1.0	Information confidentiality	X	X	X	X

EdgeLock SE050 supports CR 4.1.0 by providing encryption cryptographic services to the IoT device. Such services can be used to protect sensitive personal information at rest. If encryption keys are stored inside the EdgeLock SE050 key store, CR 1.5.1 is fulfilled and certified. In fact, EdgeLock SE050 ensures that long-lived credentials are kept safe during the device lifecycle, even if an attacker has physical access to the device. Cryptographic operations are always performed inside EdgeLock SE050 with the keys remaining in the secure environment.

4.5 SP5: Secure Provisioning and Decommissioning

The security primitive *Secure provisioning and decommissioning* is related with the process of generating and injecting key material that can be trusted by the OEM in the IoT device. This key material might include public keys or hashes to identify and validate future updates, keys and certificates to validate the cloud backend identity, secrets for encrypted connections, or device identifiers. Similarly, decommissioning describes the reverse process, where sensitive data is securely removed from the IoT device once end-of-life of the device is reached.

EdgeLock SE050 is pre-provisioned for ease of use in NXP's secure facilities with a set of device-unique keys, certificates and identifiers. Customers are therefore not required to inject additional credentials. Pre-provisioned credentials can be used to support the main use cases, including device-to-device authentication and cloud onboarding. In case the OEM needs to provision additional or different credentials than the ones securely provisioned by NXP, those can be manually created and injected in EdgeLock SE050. The provisioning of custom credentials in EdgeLock SE050 can be performed in the secure facilities of the device manufacturer or directly in the field using well-established, secure processes and protocols. You can refer to the Cryptographic Key Generation and Injection security primitive for more information on how EdgeLock SE050 supports generation and injection of custom credentials.

EdgeLock SE050 allows you to securely decommission your IoT device. Thanks to its strong tamper-resistance capabilities, EdgeLock SE050 protects keys from extraction even after the device has been decommissioned. Moreover, policies can be set to restrict or disable the usage of stored credentials. For additional security, EdgeLock SE050 supports explicit delete of created credentials (excluding some pre-provisioned credentials).

The EdgeLock SE050 Plug&Trust middleware API supports the generation and handling of key material and objects. The main EdgeLock SE050 Plug&Trust middleware API functions supporting these functionalities are listed below:

- **Get handle of (pre)provisioned keys or objects:** sss_se05x_key_object_get_handle (), sss_key_store_get_key ()
- **Generate / inject keys or objects:** sss_se05x_key_store_generate_key(), sss_se05x_key_store_set_key ()
- **Update keys or objects (including associated policies):** sss_se05x_key_store_set_key ()
- **Delete provisioned objects:** sss_key_store_erase_key (), Se05x_API_DeleteCryptoObject(), Se05x_API_DeleteSecureObject(), Se05x_API_DeleteAll()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that can be useful for the implementation of provisioning and decommissioning processes. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Symmetric AES Encryption Example (key generation):**: \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECC Signing Example (key generation):** \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA Signing Example (key generation):** \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Using policies for secure objects demo**: \simw-top\demos\se05x\se05x_policy (see Section 5.17 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging pre-provisioned and injected keys of EdgeLock SE050 and functions to securely delete provisioned credentials aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Section 4.5 at the highest security levels:

Table 8. ISA/IEC 62443-4-2 requirements supported by SP5 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
EDR/NDR 3.12	Provisioning product supplier roots of trust	-	X	X	X
EDR/NDR 3.13	Provisioning asset owner roots of trust	-	X	X	X
CR 4.2.0	Information persistence	-	X	X	X

EdgeLock SE050 supports the ISA/IEC 62443 requirements EDR/NDR 3.12 and EDR/NDR 3.13 by providing pre-provisioned credentials injected in NXP's secure facilities. Such credentials can be used as the root of trust to support a wide variety of use cases. EdgeLock SE050 also allows the customer to provision a custom root of trust. Additionally, EdgeLock SE050 helps achieving CR 4.2.0 since it prevents by design the extraction of private data, such as private keys, stored inside the SE. It also allows the user to erase the data that has been created or to set policies to restrict or disable access to stored data.

4.6 SP6: Cryptographic random number generation

The *Cryptographic random number generation* security primitive clusters those features that are related with the secure generation of random numbers. Random numbers are typically used in the context of secure protocols and related cryptographic functionalities. This primitive also includes features for the generation of true random numbers.

EdgeLock SE050 supports the generation of variable-length random numbers through the built-in AIS20 compliant Pseudo Random Number Generator (PRNG) with DRG.3 generation capabilities. The PRNG works on top of EdgeLock SE050 True Random Number Generator (TRNG) compliant to AIS31 class PTG.2. More information on EdgeLock SE050 PRNG and TRNG can be found in EdgeLock SE050 Data Sheet. Random numbers generated by EdgeLock SE050 are cryptographically secure and can be used in the context of security protocols, typically as part of broader cryptographic functionalities requiring random initialization data or IDs. For example, most secure communication protocols require the generation of a random seed or nonce, for instance, for proof of possession of the private key by the communication partner. You can refer to the Secure Communication Protocols security primitive for more information.

The EdgeLock SE050 Plug&Trust middleware API can be used to integrate this primitive in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Generate a random number**: sss_se05x_rng_get_random ()
- **Generate random data for TLS handshake**: Se05x_API_TLSGenerateRandom ()

Leveraging the random number generation capabilities provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 9 at the highest security levels.

Table 9. ISA/IEC 62443-4-2 requirements supported by SP6 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 2.12.0	Non-repudiation	X	X	X	X
CR 3.1.0	Communication integrity	X	X	X	X
CR 3.1.1	Communication authentication	-	X	X	X
CR 4.3.0	Use of cryptography	X	X	X	X

EdgeLock SE050 supports CR 4.3.0 since it provides an implementation of all common cryptographic algorithms (symmetric and asymmetric) and cryptographic functions, including functions to generate random numbers suitable for use in cryptographically secure protocols, e.g. for the generation of random session IDs or nonces. In this context, EdgeLock SE050 helps achieving CR 3.1.0 and CR 3.1.1 as described in the Secure Communication Protocols security primitive. EdgeLock SE050 helps achieving CR 2.12.0 by supporting digital signature algorithms and secure storage of certificates that are the base of all secure non-repudiation strategies.

4.7 SP7: Root of Trust

The security primitive *Root of Trust* clusters those features related with the secure establishment of the initial Root of Trust (RoT) on the security component when the device is manufactured. This might be achieved, for instance, by manufacturing the IoT device inside trusted manufacturing facilities, or by using pre-provisioned Secure Elements.

EdgeLock SE050 is pre-provisioned for ease of use in NXP's secure facilities with a set of device-unique key-pairs, certificates and identifiers that can be used to establish the initial RoT of the IoT device. Pre-provisioned credentials can be used to support the main use cases, including device-to-device authentication and onboarding to cloud backend services. In case the OEM needs to provision different credentials than the ones securely provisioned by NXP, those can be manually created and injected in EdgeLock SE050. The provisioning of custom credentials in EdgeLock SE050 can be performed in the secure facilities of the device manufacturer or directly in the field using well-established, secure processes and protocols. You can refer to Secure Provisioning and Decommissioning security primitive for more information.

Leveraging pre-provisioned or injected keys of EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 10 at the highest security levels:

Table 10. ISA/IEC 62443-4-2 requirements supported by SP7 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
EDR 3.12	Provisioning product supplier roots of trust	-	X	X	X
EDR 3.13	Provisioning asset owner roots of trust	-	X	X	X

EdgeLock SE050 supports the ISA/IEC 62443 requirements EDR 3.12 and EDR 3.13 by providing pre-provisioned credentials injected in NXP's secure facilities. Additionally, EdgeLock SE050 supports EDR 3.13 by allowing customers to easily provision their own custom root of trust in case they have their own secure programming facilities.

4.8 SP8: Secure Communication Protocols

The security primitive *Secure Communication Protocols* clusters those features that allow IoT devices to communicate securely with each other and/or the cloud backend. This primitive groups support for secure communication as well as related communication protocol support. Examples for such communication could be high-level protocols such as OPC Unified Architecture (OPC-UA) or Hypertext transfer protocol (HTTP) secured with Transport Layer Security (TLS).

EdgeLock SE050 can be leveraged to offload communication protocols cryptographic operations while keeping the cryptographic keys secure inside the SE. EdgeLock SE050 has built-in support for the widely used TLS protocol to secure upper layer communication protocols such as OPC-UA, HTTP and MQTT. EdgeLock SE050 supports the TLS protocol by protecting key-pairs and certificates that are used to authenticate the IoT device to other parties. Moreover, the EdgeLock SE050 Plug&Trust middleware provides functions to simplify the implementation of the TLS handshake by using EdgeLock SE050 cryptographic functions. More information regarding EdgeLock SE050 TLS support can be found in EdgeLock SE050 for secure connection to OEM cloud.

EdgeLock SE050 also natively supports Secure Channel Protocol (SCP), including GlobalPlatform SCP03 and FastSCP. The SCP protocol allows you to protect the integrity of local (Platform SCP) and end-to-end (Applet level SCP) communication with EdgeLock SE050. Other secure communication protocols can be supported by using EdgeLock SE050 cryptographic functions. Those include asymmetric cryptography (RSA, ECC), symmetric cryptography (AES, DES) , MAC and digest functions (HMAC, CMAC, SHA) and key agreement and derivation functions.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Perform TLS handshake:** Se05x_API_TLSCalculatePreMasterSecret (), Se05x_API_TLSGenerateRandom (), Se05x_API_TLSPerformPRF ()
- **Establish a secure SCP channel:** Se05x_API_CreateSession (), Se05x_API_SetPlatformSCPRequest ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Secure Communication Protocols* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **OPC UA Demo:**: \simw-top\demos\opc_ua (see Section 5.13 of EdgeLock SE050 Plug&Trust middleware documentation)
- **TLS Client example**: \simw-top\demos\linux\tls_client (see Section 5.12 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging the built-in support for common secure communication protocols and the cryptographic capabilities of EdgeLock SE050 aids in achieving the ISA/IEC 62443-4-2 requirements listed in Table 11 to the highest security level.

Table 11. ISA/IEC 62443-4-2 requirements supported by SP8 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.8.0	Public key infrastructure certificates	-	X	X	X
CR 3.1.0	Communication integrity	X	X	X	X
CR 3.1.1	Communication authentication	-	X	X	X
CR 3.8.0	Session integrity	-	X	X	X
CR 4.3.0	Use of cryptography	X	X	X	X

EdgeLock SE050 supports CR 4.3.0 since it provides an out-of-the-box implementation of all common cryptographic functions and cryptographic algorithms, including symmetric and asymmetric cryptography. Such functions can be used as building blocks for the implementation of secure communication protocols. EdgeLock SE050 helps achieving CR 3.1.0, CR 3.1.1 and CR 3.8.0 since it allows the user to offload the cryptographic operations that are necessary to establish the secure communication channel. Finally, EdgeLock SE050 supports CR 1.8.0 by providing a secure hardware that can securely store PKI certificates. Certificates can be stored in EdgeLock SE050 and used as part of secure communication protocols.

4.9 SP9: Secure Initialization

The *Secure Initialization* security primitive clusters features that help ensuring the authenticity and integrity of the device boot loader, firmware, and other software during the boot process. If required, the implementation may handle encrypted boot code. Depending on the use case, secure initialization might encompass one or several boot stages that are each cryptographically secured. Secure initialization may also include the validation of an application before running it on top of the platform.

EdgeLock SE050 supports the storage of public keys (RSA, ECC) that can be used to verify a signed digest of a software component before it is loaded and executed. In this way only applications that have been signed by the OEM with the corresponding private key will be accepted as valid by the system. This feature can be used to check the authenticity and integrity of boot loaders, firmware and OS applications before they are executed. For more information on how EdgeLock SE050 can be leveraged to support a secure boot, you can refer to Secure Binding for EdgeLock SE050 Application Note.

For additional protection, boot loaders and applications containing sensitive data can also be encrypted beforehand and private keys used for decryption can be stored in EdgeLock SE050. EdgeLock SE050 provides both asymmetric encryption algorithms (RSA, ECC) and symmetric encryption algorithms (AES, DES) for this purpose. EdgeLock SE050 also supports Platform Configuration Registers (PCRs): 32-byte arrays that hold the value of a SHA-256 hash. Once the value of a PCR register is updated, the new value depends on the old value and the new measure. This can be used, for instance, to enforce an integrity check and an order of execution on a chain of boot loaders.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the customers IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Verify the signature of an application digest:** sss_se05x_asymmetric_verify_digest (), sss_se05x_asymmetric_verify ()
- **Encrypt / decrypt application data:** sss_se05x_asymmetric_encrypt(), sss_se05x_asymmetric_decrypt(), sss_se05x_cipher_one_go()
- **Generate a digest of an application:** sss_se05x_digest_one_go ()
- **Write to a PCR register:** Se05x_API_WritePCR ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Secure Initialization* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **ECC Signing Example:** \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA Signing Example:** \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Message Digest Example:** \simw-top\sss\ex\md (see Section 5.2.6 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging cryptographic signature capabilities provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 12 at the highest security levels.

Table 12. ISA/IEC 62443-4-2 requirements supported by SP9 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 3.4.0	Software and information integrity	X	X	X	X
EDR/NDR 3.10.1	Update authenticity and integrity	-	X	X	X
EDR/NDR 3.14.0	Integrity of boot process	X	X	X	X
EDR/NDR 3.14.1	Authenticity of the boot process	-	X	X	X

EdgeLock SE050 supports EDR/NDR 3.14.1 and EDR/NDR 3.10.1 since it provides a secure environment to store public keys that can be used to verify the signature of applications, boot loaders and update packages before they are loaded and executed. EdgeLock SE050 supports CR 3.4.0 and EDR/NDR 3.14.0 by providing cryptographic hash functions that can be used to compute the digests of software applications that are going to be executed and compare them with pre-computed, signed digests to check integrity.

4.10 SP10: System Event Logging

The *System event logging* security primitive clusters those functionalities related to securely logging system events in an integrity protected way, with related data stored in a secure encrypted storage. This primitive also includes functionalities that can be used to implement non-repudiation strategies.

EdgeLock SE050 supports asymmetric cryptographic functions (RSA, ECC) that can be used to sign the hash of the logs before storage. If the IoT device application implements user management and authorization, user authenticators (private-public key pairs) can be securely stored in the SE and used to sign the logs generated by a specific user. This might help in the implementation of non-repudiation strategies. EdgeLock SE050 also supports hash functions (SHA-224, SHA-256, SHA-384, SHA-512) that can be used to generate and store the hash of the system logs at regular intervals.

The EdgeLock SE050 Plug&Trust middleware API can be used to integrate this primitive in the industrial IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Sign / verify logs:** sss_se05x_asymmetric_sign_digest (), sss_se05x_asymmetric_verify_digest (), sss_se05x_asymmetric_sign (), sss_se05x_asymmetric_verify ()
- **Generate hash for logs**: sss_se05x_digest_one_go ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *System Event Logging* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Message hash example**: \simw-top\sss\ex\md (see Section 5.2.6 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECC signing example**: \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA signing example**: \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging hashing and signing capabilities provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 13 at the highest security levels.

Table 13. ISA/IEC 62443-4-2 requirements supported by SP10 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 2.12.0	Non-repudiation	X	X	X	X
CR 2.12.1	Non-repudiation for all users	-	-	-	X
CR 3.9.0	Protection of audit information	-	X	X	X

EdgeLock SE050 helps achieving CR 3.9.0 and CR 2.12.0 by providing cryptographic hash and signature functions that can be used to verify the integrity and authenticity of the audit information generated by the IoT system, including system logs. In addition, if proper user authorization is enforced at the application level, EdgeLock SE050 can help to achieve CR 2.12.1 by signing logs generated by users with the user credentials securely stored in the SE.

4.11 SP11: Secure Encrypted Storage

The security primitive *Secure Encrypted Storage* clusters those features that allow the user to securely store data and maintain its integrity. It also includes features to protect the data confidentiality.

EdgeLock SE050 provides a secure, tamper-resistant hardware secure memory for the secure storage of device credentials such as keys, identifiers and certificates. EdgeLock SE050 also supports both asymmetric (RSA, ECC) and symmetric (DES, AES) cryptographic algorithms that can be used to encrypt data at rest in the IoT device. EdgeLock SE050 stores encryption keys inside its tamper-resistant secure enclave. Even if the IoT device is decommissioned, private credentials cannot be extracted from the SE. You can refer to the Secure Provisioning and Decommissioning security primitive for more information.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in your IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Encrypt/decrypt data at rest:** sss_se05x_asymmetric_encrypt (), sss_se05x_asymmetric_decrypt (), sss_se05x_cipher_one_go ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Secure (Encrypted) Storage* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging the encryption and tamper resistance capabilities of EdgeLock SE050 aids in achieving the ISA/IEC 62443-4-2 requirements listed in Table 14 to the highest security level.

Table 14. Requirements supported by SP11 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 4.1.0	Information confidentiality	X	X	X	X
CR 4.2.0	Information persistence	-	X	X	X
CR 4.3.0	Use of cryptography	X	X	X	X

EdgeLock SE050 supports CR 4.1.0 by providing a secure, tamper-resistant hardware secure memory for storing sensitive credentials such as keys and identifiers. Moreover, symmetric and asymmetric key credentials can be used to encrypt data at rest in the IoT device using any of the supported encryption algorithms. CR 4.3.0 is therefore also accomplished. EdgeLock SE050 also helps achieving CR 4.2.0 since stored data and credentials can be deleted or disabled before disposing of the IoT device as described in Secure Provisioning and Decommissioning security primitive.

4.12 SP12: Cryptographic Key Generation and Injection

The *cryptographic key generation and injection* security primitive clusters those features that allow the user to securely generate cryptographic keys and, optionally, to securely inject or import them into the IoT device. The implementation may include key exchange and key agreement support, as well as key derivation schemes.

EdgeLock SE050 natively supports the generation of symmetric (AES, DES) and asymmetric keys (ECC, RSA) directly inside the tamper-resistant, secure environment provided by the SE. Private keys will never leave the boundaries of the SE. Optionally, pre-existing keys can also be injected in EdgeLock SE050. EdgeLock SE050 also supports key exchange and key agreement algorithms (ECDH, ECDHE) and key derivation functions (HKDF, PBKDF, Wi-Fi KDF, OPC_UA KDF, PRF) that can be used, for instance, to generate session keys.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Key creation:** sss_se05x_key_store_generate_key ()
- **Key import / injection:** sss_key_store_set_key(), Se05x_API_WriteECKey (), Se05x_API_WriteRSAKey (), Se05x_API_WriteSymmKey ()
- **Key agreement:** sss_derive_key_dh ()
- **Key derivation:** , sss_derive_key_one_go ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Cryptographic Key Generation and Injection* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Symmetric AES Encryption Example (key generation):**: \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECC Signing Example (key generation):** \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA Signing Example (key generation):** \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECDH Key Derivation Example:** \simw-top\sss\ex\ecdh (see Section 5.2.7 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging key generation and key derivation capabilities provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 15 at the highest security levels.

Table 15. Requirements supported by SP12 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.5.0	Authenticator management	X	X	X	X
CR 1.5.1	Hardware security for authenticators	-	-	X	X
CR 1.8.0	Public key infrastructure certificates	-	X	X	X
CR 4.3.0	Use of cryptography	X	X	X	X

EdgeLock SE050 supports CR 1.5.0 and CR 1.5.1 by providing a secure, tamper-resistant hardware in which keys can be securely generated or injected. EdgeLock SE050 also helps achieving CR 1.8.0 by providing a secure storage for public keys and certificates. Finally, EdgeLock SE050 supports CR 4.3.0 since it provides cryptographically secure key generation capabilities and cryptographic algorithms for key agreement and key derivation.

4.13 SP13: Cryptographic Key and Certificate Store

The security primitive *Cryptographic Key and Certificate Store* clusters those features that allow the user to store key material such as keys and certificates and enforce policies on them. The key and certificate store shall provide management functionality for the key material such as policy management or key material deletion.

EdgeLock SE050 allows you to securely generate and store credentials as secure objects inside its secure tamper-resistant hardware. Cryptographic operations involving secure objects are always performed inside the SE protected environment using the built-in cryptographic functions and algorithms provided by EdgeLock SE050 applet. Key generation capabilities are covered in Cryptographic Key Generation and Injection security primitive. EdgeLock SE050 also supports access management to credentials in the form of policies that can be used to specify the operations allowed on a given credential. EdgeLock SE050 policies can be used, for example, to define if a key can be used for encryption, for signing or both and if a key is read-only or if it can be exported or deleted.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Set policy upon object creation / injection:** sss_se05x_key_store_generate_key (), sss_se05x_key_store_set_key ()
- **Update policy upon object update:** sss_se05x_key_store_set_key ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Cryptographic Key and Certificate Store* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Using policies for secure objects demo**: \simw-top\demos\se05x\se05x_policy (see Section 5.17 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging tamper resistance capabilities and key management functions provided by EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 16 at the highest security levels.

Table 16. ISA/IEC 62443-4-2 requirements supported by SP13 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.5.1	Hardware security for authenticators	-	-	X	X
CR 1.9.1	Hardware security for public key-based authentication	-	-	X	X
CR 1.14.1	Hardware security for symmetric key-based authentication	-	-	X	X

EdgeLock SE050 inherently supports CR 1.5.1, CR 1.9.1 and CR 1.14.1 since it provides a certified hardware with strong tamper-resistant protection for keys stored in the SE. EdgeLock SE050 supports both symmetric and asymmetric keys. EdgeLock SE050 also comes with advanced key management functionalities that allow the user to set policies on key objects to restrict the set of permitted operations on them.

4.14 SP14: Cryptographic Operation

The *Cryptographic Operation* security primitive clusters those features related with cryptographic functionality such as encryption, decryption, hashing, or signing. Cryptographic operation may include higher-level functionality such as certificate verification, certificate signing, and Certificate Signing Request (CSR) handling.

EdgeLock SE050 is the ideal component to support this primitive since it allows the user to securely generate and store keys in a protected tamper-resistant environment and perform cryptographic operations with stored keys using the latest, most secure cryptographic algorithms. EdgeLock SE050 supports symmetric encryption algorithms (DES, AES), public-key encryption algorithms using either RSA or ECC with support for NIST, Brainpool, Edwards and Montgomery curves, public-key signing algorithms (RSA, ECDSA, ECDAA, EdDSA), key agreement algorithms (ECDH, ECDHE) and hashing and MAC algorithms (SHA, HMAC, CMAC). For a detailed list of supported algorithms you can refer to EdgeLock SE050 Data Sheet.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go()
- **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot()
- **MAC operations:** sss_se05x_mac_one_go ()
- **Sign and verify operations:** sss_se05x_asymmetric_sign_digest (), sss_se05x_asymmetric_verify_digest (), sss_se05x_asymmetric_sign (), sss_se05x_asymmetric_verify (), Se05x_API_RSASign(), Se05x_API_ECDSASign(), Se05x_API_EdDSASign()
- **Derive key operations**: sss_se05x_derive_key_go ()
- **Key agreement operations**: sss_se05x_derive_key_dh ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Cryptographic Operation* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric (see Section 5.2.3 of EdgeLock SE050 Plug&Trust middleware documentation)
- **Message Digest Example:** \simw-top\sss\ex\md (see Section 5.2.5 of EdgeLock SE050 Plug&Trust middleware documentation)
- **HMAC Example:** \simw-top\sss\ex\hmac (see Section 5.2.6 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECC Signing Example:** \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA Signing Example:** \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECDH Key Derivation Example:** \simw-top\sss\ex\ecdh (see Section 5.2.7 of EdgeLock SE050 Plug&Trust middleware documentation)

EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 17 at the highest security levels.

Table 17. ISA/IEC 62443-4-2 requirements supported by SP14 and benefitting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.8.0	Public key infrastructure certificates	-	X	X	X
CR 1.9.0	Strength of public key-based authentication	-	X	X	X
CR 1.14.0	Strength of symmetric key-based authentication	-	X	X	X
CR 3.1.0	Communication integrity	X	X	X	X
CR 3.1.1	Communication authentication	-	X	X	X
CR 3.4.0	Software and information integrity	X	X	X	X
CR 3.4.1	Authenticity of software and information	-	X	X	X
CR 3.8.0	Session integrity	-	X	X	X
CR 3.9.0	Protection of audit information	-	X	X	X
CR 3.14.0	Integrity of boot process	X	X	X	X
CR 3.14.1	Authenticity of boot process	-	X	X	X
CR 4.1.0	Information confidentiality	X	X	X	X
CR 4.3.0	Use of cryptography	X	X	X	X
CR 7.3.1	Backup integrity verification	-	X	X	X

EdgeLock SE050 supports CR 4.3.0 by implementing all the common cryptographic algorithms for encryption, signing, hashing, key agreement and key derivation. Cryptographic functions can be applied on keys securely stored in the tamper-resistant hardware of EdgeLock SE050. This allows the user to easily achieve CR 1.9.0 and CR 1.14.0. EdgeLock SE050 cryptographic capabilities help achieving many ISA/IEC 62443-4-2 requirements: CR 3.1.0, CR 3.1.1 and CR 3.8.0 for communication security (see Secure Communication Protocols security primitive), CR 3.4.0, CR 3.4.1, CR 3.14.0 and 3.14.1 for secure boot and initialization (see Secure Initialization security primitive), CR 3.9.0 for protection of audit information (see System Event Logging security primitive), CR 4.1.0 for protection of personal information (see Protection of Personal Information security primitive), CR 7.3.1 for secure backups (see Secure Backup and Recovery security primitive) and CR 1.8.0 for secure storage of public-key certificates.

4.15 SP15: Secure Onboarding and Offboarding

The security primitive *Secure Onboarding and Offboarding* clusters those features that allow IoT devices to authenticate and connect to a local network or to a cloud backend. The IoT device identity should be unique, verifiable and trustworthy so that device registration attempts and any data uploaded to a cloud service can be trusted by the OEM. Usually the cloud backend verifies the device identity using PKI cryptography. Offboarding is the reverse process where the device is released from the network. This may be triggered prior to a secure decommissioning.

EdgeLock SE050 is pre-provisioned with device-unique key-pairs and certificates that can be used for certificate-based device authentication in the cloud onboarding process. Pre-provisioned credentials can then be used to securely establish a mutually authenticated, encrypted connection to the cloud using the TLS protocol as discussed in Secure Communication Protocols security primitive.

The EdgeLock SE050 Plug&Trust middleware provides a set of demos and code examples that help in implementing cloud onboarding in all the major cloud service platforms, including AWS, Google, Microsoft and IBM. More information on cloud service examples and how to run them is provided in Section 5.1.3 of the EdgeLock SE050 Plug&Trust middleware documentation and in the corresponding application notes (AWS IoT Core, Google Cloud Platform, Microsoft Azure IoT Hub and IBM Watson IoT).

Leveraging pre-provisioned keys and certificates in EdgeLock SE050 aids in achieving the ISA/IEC 62443-4-2 requirements listed in Table 18 to the highest security level.

Table 18. ISA/IEC 62443-4-2 requirements supported by SP15 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
EDR/NDR 3.12	Provisioning product supplier roots of trust	-	X	X	X
EDR/NDR 3.13	Provisioning asset owner roots of trust	-	X	X	X

EdgeLock SE050 supports the ISA/IEC 62443 requirements EDR/NDR 3.12 and EDR/NDR 3.13 by providing pre-provisioned keys and certificates that can be used for cloud onboarding in all major cloud platforms.

4.16 SP16: Secure Updates

The *Secure Updates* security primitive clusters functionalities to securely update an IoT device in the field. This might encompass updates and patches of firmware, software, applications, and/or the operating system. Secure updates require cryptographic functionality to verify their integrity and authenticity. If updates are downloaded from the cloud, a secure communication shall be established beforehand.

EdgeLock SE050 can be used to safely store public keys and certificates that can be used to verify the authenticity of an update before it is executed. This can be achieved using signature algorithms supported by EdgeLock SE050 to verify the signed hash of an update package. The integrity of the update can then be verified by comparing the signed hash of the update with the actual hash of the update package. EdgeLock SE050 supports all the common hash algorithms, including SHA, for this purpose. EdgeLock SE050 can also be leveraged to establish a secure TLS channel with the cloud backend to securely download updates. More information on EdgeLock SE050 secure communication protocols features can be found in the Secure Communication Protocols security primitive.

The EdgeLock SE050 Plug&Trust middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main EdgeLock SE050 Plug&Trust middleware API functions supporting the core use cases of this security primitive are listed below:

- **Verify signature of an update:** sss_se05x_asymmetric_verify_digest (), sss_se05x_asymmetric_verify ()
- **Generate digest of an update:** sss_se05x_digest_one_go ()

The EdgeLock SE050 Plug&Trust middleware also provides a set of demos and code examples that might be useful to implement the use cases supported by the *Secure Updates* security primitive. The relevant examples, along with their location in EdgeLock SE050 Plug&Trust middleware folder structure, are shown below:

- **Message Digest Example:** \simw-top\sss\ex\md (see Section 5.2.5 of EdgeLock SE050 Plug&Trust middleware documentation)
- **ECC Signing Example:** \simw-top\sss\ex\ecc (see Section 5.2.1 of EdgeLock SE050 Plug&Trust middleware documentation)
- **RSA Signing Example:** \simw-top\sss\ex\rsa (see Section 5.2.2 of EdgeLock SE050 Plug&Trust middleware documentation)

Leveraging hashing and signing capabilities of EdgeLock SE050 aids in achieving ISA/IEC 62443 4-2 compliance for the requirements listed in Table 19 at the highest security levels.

Table 19. ISA/IEC 62443-4-2 requirements supported by SP16 and benefiting from EdgeLock SE050

Code	Requirement	SL1	SL2	SL3	SL4
EDR/NDR 2.4.1	Mobile code authenticity check	-	X	X	X
CR 3.4.0	Software and information integrity	X	X	X	X
CR 3.4.1	Authenticity of software and information	-	X	X	X
CR 3.4.2	Automated notification of integrity violations	-	-	X	X
EDR/NDR 3.10.1	Update authenticity and integrity	-	X	X	X
EDR/NDR 3.12.0	Provisioning product supplier roots of trust	-	X	X	X
EDR/NDR 3.13.0	Provisioning asset owner roots of trust	-	X	X	X

EdgeLock SE050 supports EDR/NDR 3.12.0 and EDR/NDR 3.13.0 since it allows the user to securely provision key-pairs and certificates that can be used to provide a root of trust for different entities involved in the management and production of the IoT device. EdgeLock SE050 also helps achieving EDR/NDR 2.4.1, CR 3.4.0, CR 3.4.1 and EDR/NDR 3.10.1 since the established root of trust can be used to verify the authenticity of software and updates before they are executed. Pre-computed, signed hashes can be used to verify the integrity of the executed software. Finally, thanks to its tamper-detection capabilities, EdgeLock SE050 can be used in combination with IoT applications to send alerts in case of tampering attempts and in this way fulfil CR 3.4.2 as described in Anomaly Detection and Reaction security primitive.

5 ISA/IEC 62443-4-2 requirements lookup table

Table 20 maps all the ISA/IEC 62443-4-2 requirements mentioned in Section 4 to the respective security primitives.

Table 20. ISA/IEC 62443-4-2 requirements and security primitives lookup table

FR	Req.	Description	Security primitives
FR1	CR 1.2.0	Software process and device identification	SP2: Device attestation
	CR 1.2.1	Unique identification and authentication	SP2: Device attestation
	CR 1.5.0	Authenticator management	SP12: Cryptographic Key Generation and Injection

FR	Req.	Description	Security primitives
	CR 1.5.1	Hardware security for authenticators	SP1: Anomaly detection and reaction SP4: Protection of personal information SP12: Cryptographic Key Generation and Injection SP13: Cryptographic Key and Certificate Store
	CR 1.8.0	Public key infrastructure certificates	SP8: Secure Communication Protocols SP12: Cryptographic Key Generation and Injection SP14: Cryptographic Operation
	CR 1.9.0	Strength of public key-based authentication	SP14: Cryptographic Operation
	CR 1.9.1	Hardware security for public key based authentication	SP1: Anomaly detection and reaction SP13: Cryptographic Key and Certificate Store
	CR 1.14.0	Strength of symmetric key based authentication	SP14: Cryptographic Operation
	CR 1.14.1	Hardware security for symmetric key based authentication	SP1: Anomaly detection and reaction SP13: Cryptographic Key and Certificate Store
FR2	NDR/SAR 2.4.1	Mobile code authenticity check	SP16: Secure Updates
	CR 2.12.0	Non-repudiation	SP6: Cryptographic random number generation SP10: System Event Logging
	CR 2.12.1	Non-repudiation for all users	SP10: System Event Logging
FR3	CR 3.1.0	Communication integrity	SP6: Cryptographic random number generation SP8: Secure Communication Protocols SP14: Cryptographic Operation
	CR 3.1.1	Communication authentication	SP6: Cryptographic random number generation SP8: Secure Communication Protocols SP14: Cryptographic Operation
	CR 3.4.0	Software and information integrity	SP9: Secure Initialization SP14: Cryptographic Operation SP16: Secure Updates
	CR 3.4.1	Authenticity of software and information	SP14: Cryptographic Operation SP16: Secure Updates
	CR 3.4.2	Automated notification of integrity violations	SP1: Anomaly detection and reaction SP16: Secure Updates
	CR 3.8.0	Session integrity	SP8: Secure Communication Protocols SP14: Cryptographic Operation
	CR 3.9.0	Protection of audit information	SP10: System Event Logging SP14: Cryptographic Operation
	EDR/NDR 3.10.1	Update authenticity and integrity	SP9: Secure Initialization SP16: Secure Updates
	EDR/NDR 3.11.0	Physical tamper resistance and detection	SP1: Anomaly detection and reaction
	EDR/NDR 3.11.1	Notification of a tampering attempt	SP1: Anomaly detection and reaction

FR	Req.	Description	Security primitives
	EDR/NDR 3.12.0	Provisioning product supplier roots of trust	SP5: Secure Provisioning and Decommissioning SP7: Root of Trust SP15: Secure Onboarding and Offboarding SP16: Secure Updates
	EDR/NDR 3.13.0	Provisioning asset owner roots of trust	SP5: Secure Provisioning and Decommissioning SP7: Root of Trust SP15: Secure Onboarding and Offboarding SP16: Secure Updates
	EDR/NDR 3.14.0	Integrity of the boot process	SP9: Secure Initialization SP14: Cryptographic Operation
	EDR/NDR 3.14.1	Authenticity of the boot process	SP9: Secure Initialization SP14: Cryptographic Operation
FR4	CR 4.1.0	Information confidentiality	SP4: Protection of personal information SP11: Secure Encrypted Storage SP14: Cryptographic Operation
	CR 4.2.0	Information persistence	SP5: Secure Provisioning and Decommissioning SP11: Secure Encrypted Storage
	CR 4.3.0	Use of cryptography	SP6: Cryptographic random number generation SP8: Secure Communication Protocols SP11: Secure Encrypted Storage SP12: Cryptographic Key Generation and Injection SP14: Cryptographic Operation
FR7	CR 7.3.1	Backup integrity verification	SP3: Secure backup and recovery SP14: Cryptographic Operation

6 Glossary

Term	Definition
AES	Advanced Encryption Standard
CR	Component Requirement
DES	Data Encryption Standard
ECC	Elliptic-curve Cryptography
ECDH	Elliptic-curve Diffie-Hellman
ECDHE	Elliptic-curve Diffie-Hellman Ephemeral
EDR	Embedded Device Requirement
FR	Foundational Requirement
HDR	Host Device Requirement
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
KDF	Key Derivation Function
MAC	Message Authentication Code

Term	Definition
MQTT	Message Queuing Telemetry Transport
NDR	Network Device Requirement
OEM	Original Equipment Manufacturer
OS	Operating System
PCR	Platform Configuration Register
PKI	Public Key Infrastructure
PRNG	Pseudo Random Number Generator
SAR	Software Application Requirement
SCP	Secure Channel Protocol
SE	Secure Element
SHA	Secure Hash Algorithm
SL	Security Level
SP	Security Primitive
TLS	Transport Layer Security
TRNG	True Random Number Generator

7 Legal information

7.1 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.

7.2 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.

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 — While NXP Semiconductors has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP Semiconductors accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

7.3 Trademarks

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

Tables

Figures

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
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 16 June 2020
Document number: 582810