AN14827

Ease ISA/IEC 62443 compliance with EdgeLock A30

Rev. 1.0 — 7 October 2025

Application note

Document information

Information	Content
Keywords	ISA/IEC 62443, Industrial security, EdgeLock A30
Abstract	This document elaborates on the use of EdgeLock A30 features to reduce implementation complexity and to fulfill the security requirements mandated by the ISA/IEC 62443-4-2 standard.

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 life cycle.

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 Authenticator (SA) such as EdgeLock A30 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 A30 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 A30 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 A30 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 consist of 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.

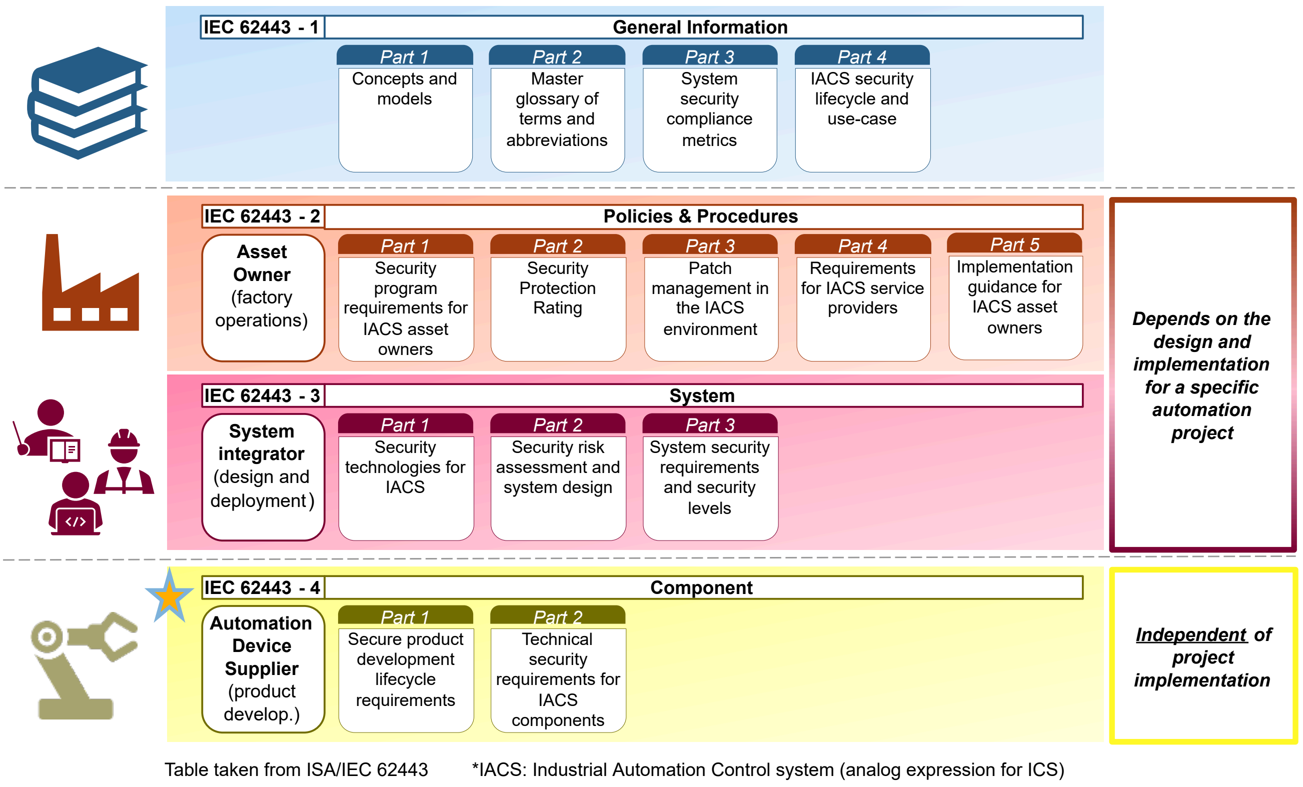
IEC 62443 - 1
General Information
Part 1
Concepts and models
Part 2
Master glossary of terms and abbreviations
Part 3
System security compliance metrics
Part 4
IACS security lifecycle and use-case
IEC 62443 - 2
Policies & Procedures
Asset Owner (factory operations)
Part 1
Security program requirements 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

Table 1. ISA/IEC 62443 security assurance levels*...continued*

Security level (SL)	Description from ISO/IEC 62443-1-1 section 10.4.3
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 A30 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 A30. The IoT device is represented by the *platform*, composed of an MCU / MPU connected to EdgeLock A30, 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.

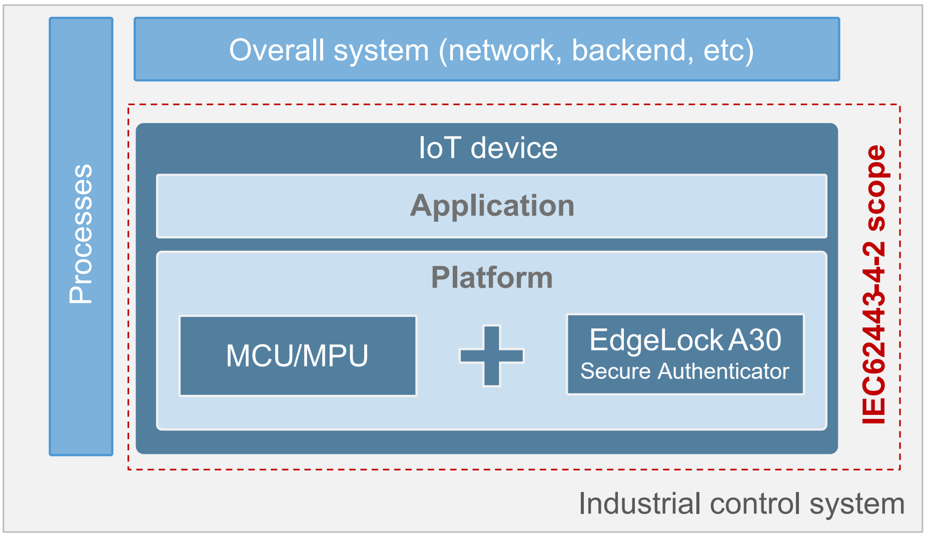

Figure 2. Breakdown of a simplified industrial control system

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 A30 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 A30 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 A30 comes with a pre-installed IoT application offering advanced key management and cryptographic functions. To ease the integration of the EdgeLock A30 functionalities in the IoT solution, EdgeLock A30 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 A30 is pre-provisioned pre-provisioned with one device-unique private key and one certificate in a highly secure and controlled environment, therefore relieving IoT device manufacturers from setting up a complex and expensive PKI infrastructure.

Note: *NXP provides commercial customization options for trust-provisioning. This allows for a customer dedicated delivery configuration. Reach out to your local sales representative for more information.*

As a result, EdgeLock A30, 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 A30 can add a valuable contribution are listed in Table 3. In the next sections, EdgeLock A30 features will be put in the context of each security primitive listed in Table 3. Which ISA/IEC 62443-4-2 requirements EdgeLock A30 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
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 life cycle state. This primitive includes logical and physical tamper detection and tamper protection of the IoT device.

EdgeLock A30 provides enhanced runtime 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. Any anomaly occurring during that stage would result in a life cycle state of the Secure Authenticator (SA) in which mission-critical keys and certificates are not available to the host device. Further details on how the EdgeLock A30 ensures that long-lived credentials are kept safe during the device life cycle are provided in Cryptographic Key and Certificate Store security primitive. In addition, EdgeLock A30 provides a tamper-resistant platform, certified level at CC EAL 6+. In case a physical tampering of the SA is detected, EdgeLock A30, 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 runtime integrity and triggering actions on detected anomalies provided by EdgeLock A30 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 A30

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 A30 inherently supports the ISA/IEC 62443 requirement EDR/NDR 3.11.0 with its integrated tamper protections certified at CC EAL 6+ including AVA_VAN 5, the highest achievable level in vulnerability analysis and penetration testing. Additionally, application developers can leverage EdgeLock A30 tamper reaction features to easily fulfill CR 3.4.2 and 3.11.1. Finally, if authenticator keys are stored inside the EdgeLock A30 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 A30 ensures that long-lived credentials are kept safe during the device life cycle, even if an attacker has physical access to the device. Cryptographic operations are always performed inside EdgeLock A30 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 life cycle state. Genuine identification requires ensuring a unique identification of the IoT device.

EdgeLock A30 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 A30 and protected against deletion and overwriting using the appropriate policies. EdgeLock A30 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 A30 is pre-provisioned with one private EC key and the corresponding certificate, that can be used to attest the device identity in different use cases, including cloud onboarding and device-to-device authentication.

Note: *NXP provides commercial customization options for trust-provisioning. This allows for a customer dedicated delivery configuration. Reach out to your local sales representative for more information.*

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the industrial IoT solution. The NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Read EdgeLock A30 pre-injected UID:** nx_GetCardUID()
  Read binary identifier / certificate: sss_key_store_get_key (), nx_ReadData()
- **Inject a binary identifier / certificate**: sss_key_store_set_key (), nx_WriteData()

The NX Middleware also provides a set of demos and code examples that might be useful to show how to use EdgeLock A30 and ease the implementation of the use cases supported by the *Device attestation* security primitive. The relevant examples are shown below:

- **Get UID example:** ex_get_uid
- **Get Version example**: ex_get_version
- **File Management example**: ex_file_mgnt

Leveraging identifier and certificate management capabilities provided by EdgeLock A30 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 A30

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 A30 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 A30 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 A30 supports ECDSA (NIST P-256 or Brainpool P256r1) signature algorithms for this purpose. The backup digests can be generated by EdgeLock A30 using supported hash functions (SHA-256/384). For additional protection of the backup data, EdgeLock A30 can be leveraged to encrypt the backup before saving it locally or uploading it to the cloud. EdgeLock A30 provides symmetric encryption algorithms (AES-128/256) for this purpose. The EdgeLock A30 tamper-resistant hardware ensures that signing keys and encryption keys are protected.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Sign / verify a digest of a backup:**sss_asymmetric_sign_digest(), sss_asymmetric_sign_one_go(), sss_asymmetric_sign_init(), sss_asymmetric_sign_update(), sss_asymmetric_sign_finish(), sss_nx_asymmetric_verify_digest(), sss_asymmetric_verify_one_go(), sss_asymmetric_verify_init(), sss_asymmetric_verify_update(), sss_asymmetric_verify_finish()
- **Generate a digest of a backup:** sss_digest_one_go(), sss_digest_one_go(), sss_digest_update(), sss_digest_finish()
- **Symmetric Encrypt / decrypt of a backup:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish()

The NX 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 NX Middleware folder structure, are shown below:

- **Message Digest example**:ex_md
- **ECC Signing/Verifing example**: ex_ecc
- **Symmetric AES Encryption/Decryption example:** ex_symmetric

Leveraging encryption and signature functions provided by EdgeLock A30 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 A30

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

EdgeLock A30 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 A30 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 A30 secure tamper-resistant hardware and never leave the boundaries of the SA.

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 A30 provides support for cryptographic algorithms that can be used to encrypt sensitive information and protect it from unauthorized access and disclosure. EdgeLock A30 supports symmetric (AES-128/256) encryption for this purpose. EdgeLock A30 implements strong hardware security for protecting encryption keys. In combination with proper user authorization implemented at the application level, EdgeLock A30 can be used to grant access to sensitive information only to authorized users.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Symmetric Encrypt sensitive personal information:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish()
- **Symmetric Decrypt sensitive personal information:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish()

The NX 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 NX Middleware folder structure, are shown below:

- **Symmetric AES Encryption/Decryption example:** ex_symmetric

EdgeLock A30 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 A30

Code	Requirement	SL1	SL2	SL3	SL4
CR 1.5.1	Hardware security for authenticators	-	-	X	X
CR 4.1.0	Information confidentiality	X	X	X	X

EdgeLock A30 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 A30 key store, CR 1.5.1 is fulfilled and certified. In fact, EdgeLock A30 ensures that long-lived credentials are kept safe during the device life cycle, even if an attacker has physical access to the device. Cryptographic operations are always performed inside EdgeLock A30 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 A30 is pre-provisioned for ease of use in NXP's secure facilities with one device-unique private EC key, certificate (containing the corresponding public key) and identifier. 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 A30. The provisioning of custom credentials in EdgeLock A30 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 A30 supports generation and injection of custom credentials.

Note: *NXP provides commercial customization options for trust-provisioning. This allows for a customer dedicated delivery configuration. Reach out to your local sales representative for more information.*

EdgeLock A30 allows you to securely decommision your IoT device. Thanks to its strong tamper-resistance capabilities, EdgeLock A30 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 A30 supports overwriting of created credentials (excluding some pre-provisioned credentials).

The NX Middleware API supports the generation and handling of key material and objects. The main NX Middleware API functions supporting these functionalities are listed below:

- **Get handle of (pre)provisioned keys or objects:** sss_key_object_get_handle(), sss_key_store_get_key()
- **Generate / inject keys or objects:** sss_key_store_get_key(), sss_key_store_set_key()
- **Update keys or objects (including associated policies):** sss_key_store_set_key()

The NX 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 NX Middleware folder structure, are shown below:

- **Symmetric AES Encryption/Decryption example:** ex_symmetric
- **ECC Signing/Verifing example (key generation)** ex_ecc

Leveraging pre-provisioned and injected keys of EdgeLock A30 and functions to overwrite 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 A30

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 A30 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 A30 also allows the customer to provision a custom root of trust. Additionally, EdgeLock A30 helps achieving CR 4.2.0 since it prevents by design the extraction of private data, such as private keys, stored inside the SA. It also allows the user to overwrite 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 A30 supports the generation of variable-length random numbers through the built-in NIST SP800-90A, AIS20 compliant Pseudo Random Number Generator (PRNG) with DRG.4 generation capabilities. The PRNG works on top of EdgeLock A30 True Random Number Generator (TRNG) compliant to NIST SP800-90B, AIS31 class PTG.2. Random numbers generated by EdgeLock A30 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 NX Middleware API can be used to integrate this primitive in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Generate a random number**: sss_rng_get_random(), nx_CryptoRequest_RNG()

Leveraging the random number generation capabilities provided by EdgeLock A30 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 A30

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 A30 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, for example, for the generation of random session IDs or nonces. In this context, EdgeLock A30 helps achieving CR 3.1.0 and CR 3.1.1 as described in the Secure Communication Protocols security primitive. EdgeLock A30 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 Authenticators.

EdgeLock A30 is pre-provisioned for ease of use in NXP's secure facilities with one device-unique private EC key, certificate (containing the corresponding public key) 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 A30. The provisioning of custom credentials in EdgeLock A30 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.

Note: *NXP provides commercial customization options for trust-provisioning. This allows for a customer dedicated delivery configuration. Reach out to your local sales representative for more information.*

Leveraging pre-provisioned or injected keys of EdgeLock A30 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 A30

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 A30 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 A30 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 A30 can be leveraged to offload communication protocols cryptographic operations while keeping the cryptographic keys secure inside the SA. EdgeLock A30 has built-in support for the widely used TLS protocol to secure upper layer communication protocols such as OPC-UA, HTTP and MQTT. EdgeLock A30 supports the TLS protocol by protecting key-pairs and certificates that are used to authenticate the IoT device to other parties.

A30 supports two protocols to establish a secure messaging channel:

- **PKI-based Asymmetric Mutual Authentication**
  - It is based on **Sigma-I 256-bit ECC** (NIST P-256 or brainpoolP256r1).
  - Generates AES-128 or 256 session keys for Sigma-I mutual authentication message exchange.
  - Generates AES-128 or 256 session keys used for EV2 secure messaging channel.
- **AES-based Symmetric Mutual Authentication**
  - The same protocol as introduced in MIFARE DESFire EV2 products.
  - Based on AES-128 or AES-256.
  - Generates AES-128 or 256 session keys for EV2 secure messaging channel.
- Both mutual authentication methods initiate a MIFARE DESFire and NTAG42x compatible **EV2 secure messaging channel** (authenticated session).
  - AES-128 or AES-256 session encryption/decryption and MAC keys.

- Access rights to subsequent commands and files granted after successful mutual authentication depending on configuration.
- A30 supports one open secure messaging channel (authenticated session) at one time.

Other secure communication protocols can be supported by using EdgeLock A30 cryptographic functions. Those include asymmetric cryptography (ECC), symmetric cryptography (AES) , MAC and digest functions (HMAC, CMAC, SHA) and key agreement and derivation functions.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Establish a secure secure messaging channe:** sss_nx_session_open()

The NX 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 NX Middleware folder structure, are shown below:

- **OpenSSL Engine/Provider: TLS Client example**
- **AWS Demo for Raspberry Pi**
- **AWS Cloud Demo on FreeRTOS**
- **MCXN-947 AWS Cloud Demo**

Leveraging the built-in support for common secure communication protocols and the cryptographic capabilities of EdgeLock A30 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 A30

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 A30 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 A30 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 A30 supports CR 1.8.0 by providing a secure hardware that can securely store PKI certificates. Certificates can be stored in EdgeLock A30 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 A30 supports the storage of public keys (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 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 A30. EdgeLock A30 provides both asymmetric encryption algorithms (ECC) and symmetric encryption algorithms (AES) for this purpose.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the customers IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Verify the signature of an application digest:** sss_asymmetric_verify_digest (), sss_asymmetric_verify_one_go(), sss_asymmetric_verify_init(), sss_asymmetric_verify_update(), sss_asymmetric_verify_finish()
- **Generate a digest of an application:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_finish
- **Symmetric Encrypt / decrypt application data:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish()

The NX 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 NX Middleware folder structure, are shown below:

- **ECC Signing/Verifing example**: ex_ecc
- **Message Digest example**:ex_md
- **Symmetric AES Encryption/Decryption example:** ex_symmetric

Leveraging cryptographic signature capabilities provided by EdgeLock A30 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 A30

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 A30 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 A30 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 A30 supports asymmetric cryptographic functions (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 SA and used to sign the logs generated by a specific user. This might help in the implementation of non-repudiation strategies. EdgeLock A30 also supports hash functions (SHA-256 and SHA-384) that can be used to generate and store the hash of the system logs at regular intervals.

The NX Middleware API can be used to integrate this primitive in the industrial IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Sign / verify logs:**sss_asymmetric_sign_digest(), sss_asymmetric_sign_one_go(), sss_asymmetric_sign_init(), sss_asymmetric_sign_update(), sss_asymmetric_sign_finish(), sss_nx_asymmetric_verify_digest(), sss_asymmetric_verify_one_go(), sss_asymmetric_verify_init(), sss_asymmetric_verify_update(), sss_asymmetric_verify_finish()
- **Generate hash for logs**: sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_finish

The NX 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 NX Middleware folder structure, are shown below:

- **ECC Signing/Verifing example**: ex_ecc
- **Message Digest example**: ex_md

Leveraging hashing and signing capabilities provided by EdgeLock A30 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 A30

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 A30 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 A30 can help to achieve CR 2.12.1 by signing logs generated by users with the user credentials securely stored in the SA.

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 A30 provides a secure, tamper-resistant hardware secure memory for the secure storage of device credentials such as keys, identifiers and certificates. EdgeLock A30 also supports both asymmetric (ECC) and symmetric (AES) cryptographic algorithms that can be used to encrypt data at rest in the IoT device. EdgeLock A30 stores encryption keys inside its tamper-resistant secure enclave. Even if the IoT device is decommissioned, private credentials cannot be extracted from the SA. You can refer to the Secure Provisioning and Decommissioning security primitive for more information.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in your IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Symmetric Encrypt/decrypt data at rest:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish()

The NX 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 NX Middleware folder structure, are shown below:

- **Symmetric AES Encryption/Decryption example:** ex_symmetric

Leveraging the encryption and tamper resistance capabilities of EdgeLock A30 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 A30

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 A30 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 A30 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 A30 natively supports the generation of symmetric (AES) and asymmetric keys (ECC) directly inside the tamper-resistant, secure environment provided by the SA. Private keys will never leave the boundaries of the SA. Optionally, pre-existing keys can also be injected in EdgeLock A30. EdgeLock A30 also supports key exchange and key agreement algorithms (ECDH, ECDHE) and HKDF (RFC5869) key derivation function that can be used, for instance, to generate session keys.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Key creation:** sss_key_store_generate_key ()
- **Key import / injection:** sss_key_store_set_key(), nx_ManageKeyPair(), nx_ChangeKey()
- **Key agreement:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2
- **Key derivation:** , sss_derive_key_one_go()

The NX 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 NX Middleware folder structure, are shown below:

- **Symmetric AES Encryption example (key injection):** ex_symmetric
- **ECC Signing example (key generation):** ex_ecc
- **ECDH Key Derivation example:** ex_ecdh

Leveraging key generation and key derivation capabilities provided by EdgeLock A30 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 A30

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 A30 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 A30 also helps achieving CR 1.8.0 by providing a secure storage for public keys and certificates. Finally, EdgeLock A30 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 A30 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 SA protected environment using the built-in cryptographic functions and algorithms provided by EdgeLock A30 application. Key generation capabilities are covered in Cryptographic Key Generation and Injection security primitive. EdgeLock A30 also supports access management to credentials in the form of a key policy that can be used to specify the operations allowed on a given credential. EdgeLock A30 policies can be used, for example, to define if an EC key can be used for SIGMA-I mutual authentication, for signing or both.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Set EC key policy upon object creation / injection:** nx_ManageKeyPair()
- **Update EC policy upon object update:** nx_ManageKeyPair()
- **Update AES key policy:** nx_ChangeKey()

The NX 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 NX Middleware folder structure, are shown below:

- **ECC Signing Example (key generation):** ex_ecc
- **Symmetric AES Encryption Example (key injection):** ex_symmetric
- **Update symmetric AES key:** ex_update_key

Leveraging tamper resistance capabilities and key management functions provided by EdgeLock A30 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 A30

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 A30 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 SA. EdgeLock A30 supports both symmetric and asymmetric keys. EdgeLock A30 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 A30 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 A30 supports symmetric encryption algorithms (AES), public-key signing algorithms using ECDSA with support for NIST and Brainpoolcurves, 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 A30 Data Sheet.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Symmetric Encrypt / decrypt application data:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update(), and sss_nx_cipher_finish()
- **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_finish
- **MAC operations:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish()
- **Sign and verify operations:**sss_asymmetric_sign_digest(), sss_asymmetric_sign_one_go(), sss_asymmetric_sign_init(), sss_asymmetric_sign_update(), sss_asymmetric_sign_finish(), sss_nx_asymmetric_verify_digest(), sss_asymmetric_verify_one_go(), sss_asymmetric_verify_init(), sss_asymmetric_verify_update(), sss_asymmetric_verify_finish()
- **Key agreement:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2
- **Key derivation:** , sss_derive_key_one_go()
- **Generate a random number**: sss_rng_get_random()

The NX 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 NX Middleware folder structure, are shown below:

- **Symmetric AES Encryption example:** ex_symmetric
- **Message Digest example**:ex_md,
- **ECC Signing/Verifing example:** ex_ecc
- **ECDH Key Derivation example:** ex_ecdh
- **HKDF Key derivation example:** ex_hkdf
- **RNG example:** ex_rng

EdgeLock A30 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 benefiting from EdgeLock A30

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

Table 17. ISA/IEC 62443-4-2 requirements supported by SP14 and benefiting from EdgeLock A30*...continued*

Code	Requirement	SL1	SL2	SL3	SL4
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 A30 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 A30. This allows the user to easily achieve CR 1.9.0 and CR 1.14.0. EdgeLock A30 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 A30 is pre-provisioned with one device-unique private EC key and the corresponding certificate 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.

Note: *NXP provides commercial customization options for trust-provisioning. This allows for a customer dedicated delivery configuration. Reach out to your local sales representative for more information.*

The NX Middleware provides a set of demos and code examples that help in implementing cloud onboarding in all the major IoT cloud service platforms, including AWS and Azure.

Leveraging pre-provisioned keys and certificates in EdgeLock A30 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 A30

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 A30 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 A30 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 A30 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 A30 supports all the common hash algorithms, including SHA, for this purpose. EdgeLock A30 can also be leveraged to establish a secure TLS channel with the cloud backend to securely download updates. More information on EdgeLock A30 secure communication protocols features can be found in the Secure Communication Protocols security primitive.

The NX Middleware API can be used to simplify the integration of the abovementioned use cases in the IoT solution. The main NX Middleware API functions supporting the core use cases of this security primitive are listed below:

- **Verify signature of an update:** sss_asymmetric_verify_digest (), sss_asymmetric_verify_one_go(), sss_asymmetric_verify_init(), sss_asymmetric_verify_update(), sss_asymmetric_verify_finish()
- **Generate digest of an update:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_finish)

The NX 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 NX Middleware folder structure, are shown below:

- **ECC Signing/Verifing example**: ex_ecc
- **Message Digest example**:ex_md

Leveraging hashing and signing capabilities of EdgeLock A30 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 A30

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

Table 19. ISA/IEC 62443-4-2 requirements supported by SP16 and benefiting from EdgeLock A30*...continued*

Code	Requirement	SL1	SL2	SL3	SL4
EDR/NDR 3.13.0	Provisioning asset owner roots of trust	-	X	X	X

EdgeLock A30 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 A30 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 A30 can be used in combination with IoT applications to send alerts in case of tampering attempts and in this way fulfill 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
	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

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

FR	Req.	Description	Security primitives
			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
	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

Table 21. Glossary

Term	Definition
AES	Advanced Encryption Standard
CR	Component Requirement
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
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
SA	Secure Authenticator
SAR	Software Application Requirement
SCP	Secure Channel Protocol
SHA	Secure Hash Algorithm
SL	Security Level
SP	Security Primitive
TLS	Transport Layer Security
TRNG	True Random Number Generator

7 Revision history

Table 22. Revision history

Document ID	Release date	Description
AN14827 v.1.0	7 October 2025	Initial version

Legal information

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.

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 https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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

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

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

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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

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

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

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

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

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

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