AN14252

Secure Medical IoT application with NXP Secure Devices

Rev. 2.0 — 5 December 2025 Application note

Document information

Information	Content
Keywords	Medical IoT, secure element, EdgeLock, SE05x, A5000, A30, EdgeLock 2GO, FDA, EU MDR
Abstract	This document describes how NXP secure solutions such as EdgeLock SE05x/A5000/A30 and EdgeLock 2GO can be integrated in medical IoT devices to meet medical cybersecurity regulations, such as FDA and EU MDR regulations.

1 Introduction to medical IoT

The Internet of Things (IoT) continues to proliferate, seamlessly integrating into diverse fields, such as industry, agriculture, smart cities, and even healthcare. **Medical IoT (MIoT)**, also known as healthcare IoT or **Internet of Medical Things (IoMT)**, is an ever-expanding industrial field that is leading to a growing ecosystem of smart, connected healthcare IoT devices: from small wearables that can track the patients health parameters, such as temperature or blood pressure, to high-end hospital equipment such as Magnetic Resonance Imaging (MRI) machines, or infusion pumps in Intensive care units. This expansion has been propelled by medical and electronic advancements, such as device miniaturization and sensor accuracy, and by patients and doctors' need to have constant access to real-time health data.

Much like other IoT devices, the focus in MIoT devices is on sensing and actuating. With the data collected from this, it is able to make smart decisions. This data can also be collected for further analysis. More often than not this collected data is sensitive. However, this data can be secured using cryptography from a secure element.

In the MIoT ecosystem, medical devices are no longer isolated entities. Devices are required to communicate and exchange information with many actors, such as other medical devices, a patient's or doctor's mobile phone or tablet, cloud servers, and other IT systems. On top of this connected infrastructure, MIoT devices are expected to constantly provide real time data, automatically detect health problems and update patients and doctors with automated notifications.

As healthcare organizations continue to embrace MIoT technology, it is of utmost importance to implement strong cybersecurity practices to build public trust, preserve patients' privacy and security, prevent unintended usage of the medical equipment and avoid disruption of critical components of healthcare infrastructures. In fact, attackers can leverage insecure communication protocols or hardware and software vulnerabilities to gain unauthorized access to patient information and medical records, manipulate patient information or even disrupt the operation of hospitals and healthcare facilities as well as critical medical equipment putting at risk the safety and health of patients under treatment. Implementing strong security countermeasures, supported by cryptographic credentials and protocols, is therefore an essential requirement to build a secure and reliable MIoT infrastructure.

1.1 Overview of FDA and EU MDR cybersecurity regulations

Regulatory agencies worldwide are increasingly recognizing the critical need to regulate IoT healthcare cybersecurity. As the healthcare industry becomes more interconnected through IoT devices, ensuring robust security measures is paramount. These regulations aim to safeguard sensitive patient data, prevent unauthorized access, and mitigate the risk of cyber threats that could compromise patient safety and privacy. Some of the markets where such regulations have been put in place are the United States (US) and the European Union (EU).

In the US, the Food and Drug Administration (FDA) released a final guidance for medical IoT devices titled Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions which contains the information that must be submitted for the premarket evaluation of products that involve cybersecurity risks. The regulation includes pre-market guidance, as well as guidance related to monitoring, identifying, and addressing cybersecurity vulnerabilities in medical devices once they are on the market.

The EU Medical Device Regulation (MDR) is the set of regulations that governs the production and distribution of medical devices in the EU. Compliance with this regulation is mandatory for medical device companies that want to market or sell their products in the European Economic Area (EEA). Similar to the FDA regulation, the MDR contains guidance on cybersecurity measures that MIoT devices must implement to be commercialized in the EU. Additionally, to the MDR, the General Data Protection Regulation (GDPR) must also be taken into account when commercializing IoT products in the EU.

Even though the above-mentioned regulations are in place, these are still in the early stages of incorporating cybersecurity concerns and it is expected that in the coming years stricter cybersecurity measures will be

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

mandated. For this reason, it is important to design medical IoT devices with future-proof security that can withstand new, stricter regulations.

1.2 Introducing NXP secure solutions for MIoT

NXP provides scalable, flexible, and secure solutions to develop future-proof MIoT solutions that fulfill the security and connectivity requirements mandated by FDA and EU MDR. Also, as MIoT is progressing, the separation between such medical devices from everyday devices is disappearing. This opens up the possibility of mandating the devices to be compliant with regulations beyond the scope of healthcare, such as GDPR.

Enabling a security level “high” on MIoT devices is as easy as integrating the NXP EdgeLock SE05x/A5000: a ready-to-use SE solution tailor-made for the IoT that provides a secure, CC EAL 6+, AVA_VAN.5 certified tamper-resistant hardware to protect mission-critical cryptographic credentials as well as a secure environment to offload cryptographic operations. EdgeLock SE05x/A5000 is pre-provisioned with keys and credentials in a highly secure and controlled environment, therefore relieving device manufacturers from setting up a complex and expensive Public Key Infrastructure (PKI). It also comes with a pre-installed applet and the EdgeLock Plug & Trust middleware package that ease the integration of the secure element in the device MCU/MPU.

The latest addition to the EdgeLock Discrete secure element family is EdgeLock SE052F, the industry’s first hardware secure element certified for the latest FIPS 140-3 standard with overall security 3. As part of the proven EdgeLock SE05x family, the EdgeLock SE052F combines the flexibility of a secure element with the newest generation of the Federal Information Processing Standard (FIPS), a U.S. and Canadian federal standard for data security required by NIST for participation in federal projects. In addition, this standard has become an indicator of advanced security capabilities. This makes it easier to design secure and differentiated devices across the Health-tech market.

The EdgeLock A30 secure authenticator complements the S05x secure element portfolio. The EdgeLock A30 is a ready-to-use, Common Criteria EAL 6+ certified secure authenticator that includes advanced security mechanisms, including AVA_VAN.5, with symmetric and asymmetric crypto. EdgeLock A30 is pre-provisioned with one device-unique private key and one certificate in a highly secure and controlled environment, therefore relieving medical device manufacturers from setting up a complex and expensive PKI infrastructure. To simplify the products process and reduce the production cost the EdgeLock A30 device UID and the application X.509 certificate can be downloaded via the EdgeLock 2Go service. This eliminates the need for medical device manufacturers to read the credential from each individual EdgeLock A30 device.

To abstract the complexity of key and certificate management in secure elements and authenticators, NXP offers **EdgeLock 2GO**: a fully managed cloud platform that allows customers to create and manage secure objects, such as symmetric roots of trust, key-pairs, and certificates, which are then securely provisioned (either remotely or locally) into the secure elements of IoT devices. This gives customers the flexibility to securely manage the credentials of MIoT devices already deployed in the field and to quickly and easily update them to meet new security requirements or react to security incidents.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

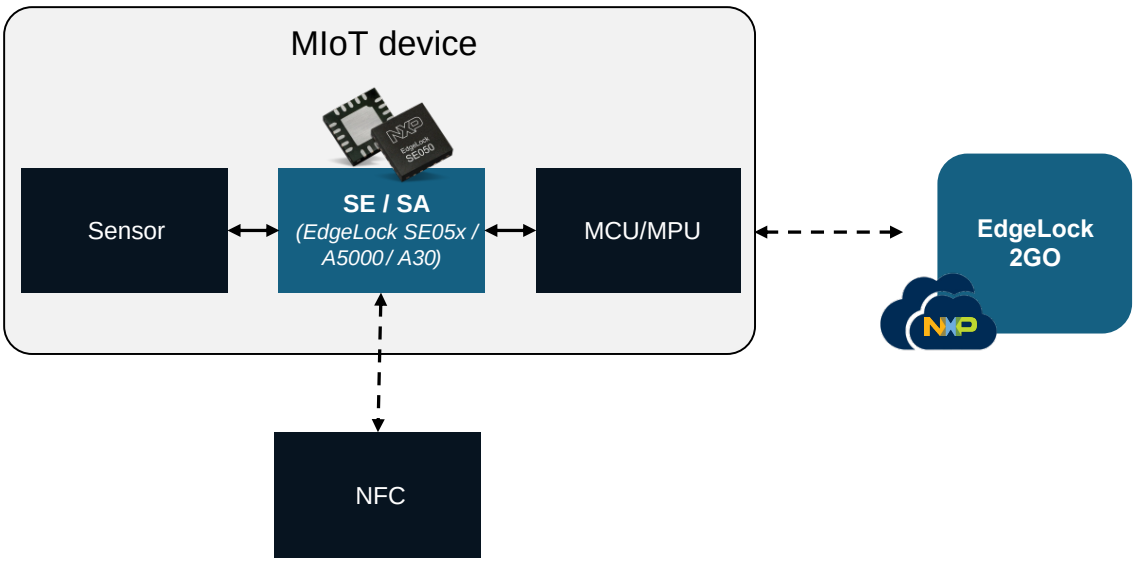

Figure 1. Example of how to integrate NXP EdgeLock secure elements and secure authenticators into an MIoT device

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

2 Architecture of an IoMT System

The IoMT infrastructure consists of several actors and components that communicate with one another with the objective of collecting healthcare data, analyzing it and distributing it to end users, such as patients, doctors, and medical personnel. The typical IoMT architecture is depicted in Figure 2.

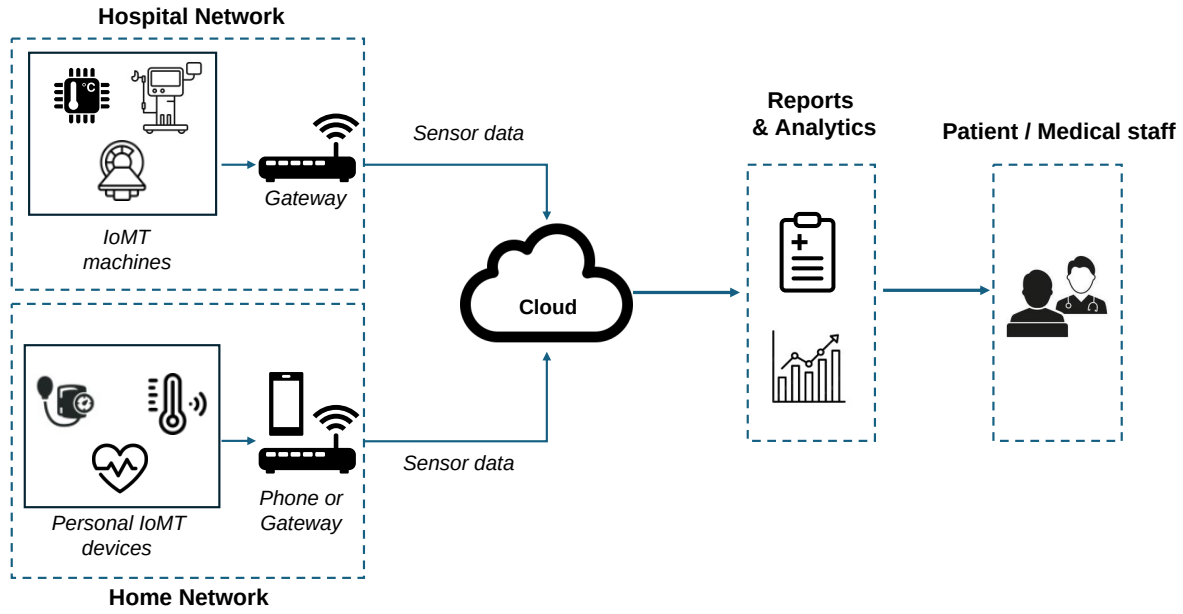

Figure 2. Architecture of a typical IoMT system

- **IoMT devices:** IoMT devices vary in shape, size, and function. These devices range from personal devices, for example, wearables to track basic health parameters (heart rate, blood pressure, temperature), to sophisticated smart medical equipment (infusion pumps, pacemakers, MRI devices, etc.). These smart medical devices generate data that can be collected locally and/or sent to a cloud infrastructure through the connectivity interface of the devices. An amount of local intelligence is present in the device. The devices must be able to adapt to the data it collects, such as having to increase the insulin being pumped out. Beyond this local intelligence there is also the possibility of remote intelligence, such an example would be if the parameters of the device need to be adapted based on collected data.
- **IoT gateway:** Individual IoMT devices might not be able to communicate directly with the cloud services themselves or even with each other. A gateway is a layer in between IoMT devices and the cloud that facilitates the communication between the two. The gateway can be a simple data forwarder or a more complex device performing tasks such as prefilter data, aggregate data, or convert data for the necessary Internet protocol. In a home network the gateway role can be enacted by a mobile phone or a PC, while in a medical facility network it can be a dedicated device with advanced features and capacity to interface with thousands of IoMT devices.
- **Cloud:** The cloud infrastructure stores and processes data collected directly from the medical devices or from the IoT gateways. It provides scalability, security, and accessibility for healthcare providers and patients.
- **Reports and analytics:** The end result of IoMT systems is to process large amounts of healthcare data and generate useful medical reports and analytics. Processed data is typically processed in the cloud and can be

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

compiled into different type of reports, which are then passed to patients or medical professional for analysis or for preventative actions

- **End users:** The people on the other end of the system who are authorized to access medical data and reports stored in the cloud. It is of utmost importance that all end users are authenticated and have the correct authorization to access data, reports, and analytics. In IoMT, typical end users are:
  - **Medical staff:** Medical staff would need to have access to current and past data, and be able to analyze the data from the reports. From here they would be able to advise patients or other staff based on the results. For example, prescribing medication or committing to further medical tests.
  - **Patients:** These end users might need to get access to reports or other collected data. Typically, these reports and data would be viewed on a personal computer or a mobile phone from the patient's home network over a potentially unsecured connection.
  - **Relatives:** Also end uses that would interact with the patients frequently, such as a partner or children. Depending on the situation, they may need to interact with certain aspects of the system or have no access.

2.1 Security considerations for IoMT

IoMT devices generate sensitive data that must be protected to ensure authenticity, confidentiality, and integrity. The following security considerations are essential to guarantee the secure operation of IoMT devices:

- **Authenticate devices:** The initial step for any device in the system is to authenticate. Only authenticated devices should be able to enter the network and verify information from other devices. Implementing robust authentication mechanisms, supported by strong cryptographic protocols, ensures that only authenticated devices can send data to the cloud and access potentially sensible information.
- **Authorization:** Within the system various entities will be connected. These entities will require access to other entities/data but not necessarily all entities/data. This could be due to various reasons such as protecting data from unwarranted access. This is where the concept of authorization comes in. It is the set of permissions an entity has; this enables it access to specific resources and locks away resources that are not required.
- **Implement secure communication with other devices and the cloud:** Medical devices in the network will be required to communicate with each other over a wireless interface and with cloud services over potentially insecure networks such as the Internet. Due to this and the private nature of the data generated by medical IoT devices, it is essential that data is properly encrypted before it is sent over the network, so only intended recipients can read the data. This can be achieved using protocols such as Transport Layer Security (TLS) and strong cryptographic credentials.
- **Authenticate transactions and ensure integrity:** IoMT devices in the IoT network will perform many transactions daily, ranging from a few messages to continuous measurements. A simple change in the value of vital metrics collected by medical devices may affect the ways in which patient care is delivered and lead to fatal consequences. Transaction undeniability of medical devices is therefore essential to prevent attackers from generating fake transactions or altering them. Digital cryptographic signatures are the key enabler to ensure the authenticity of transactions: by using a device-unique private key securely stored in the IoMT device, the device can sign transactions that can be later verified by any third-party system using the associated public key.
- **Protection of data at rest:** In certain scenarios sensitive data will be stored locally on the device. This could be due to the network connections being disrupted, or the requirement for data to be preprocessed in the device itself. This stored data would need to be protected; on top of implementing access through authentication and authorization, there should also be confidentiality to the data. This would include a method of encrypting data before it is stored.
- **Device attestation:** There are multiple processes to creating the end product. This includes but is not limited to the placement of components onto the PCB, the flashing of firmware, and the validation of the device. When the final device is created and received, it is important to ensure that the device is genuine and has not been tampered with. This is the concept of device attestation. It is achieved by ensuring the device is

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

cryptographically signed. This can also help protect the recipient of the device against receiving counterfeit and cloned products.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

3 Regulations for IoMT devices

This section will provide a brief overview of the medical regulations that are covered in this document:

- The **FDA regulation** is described in Section 3.1. The FDA regulation covers medical device requirements for the US market.
- The **MDR regulation** is described in Section 3.2. The MDR regulation covers medical device requirements for the EU market.

3.1 FDA regulation

Updated in 2025—superseding the September 2023 (and 2014) versions—the FDA's final guidance "*Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions*" defines what must be submitted for devices with cybersecurity risks. It retains its emphasis on premarket design, lifecycle monitoring, and vulnerability management, but now establishes statutory requirements (under FD&C Act § 524B), including integrated design controls and ongoing SBOMs.

Although still nonbinding, the 2025 guidance makes compliance with § 524B a prerequisite for FDA to accept and authorize "cyber devices"—failure to maintain cybersecurity processes may lead to Refuse-to-Accept (RTA) and enforcement under the False Claims Act.

The three main sections of the 2025 FDA Guidance:

1. **General Principles**
   Confirms cybersecurity is part of device safety and quality, reinforcing integration with Quality System Regulation (21 CFR 820).
   Encourages Secure-by-Design via a Secure Product Development Framework (SPDF), now tied explicitly to design controls and lifecycle practices.
   Clarifies "cyber device" more broadly: any device with software that can connect (directly or indirectly), regardless of network connectivity.
2. **SPDF & Risk Management (Premarket → Postmarket)**
   Strengthens mandates around threat modeling, secure coding, CAPA, and encryption—integrated from development onward.
   Requires Software Bill of Materials (SBOM) for all cyber devices—with transparency on origin, support timelines, and vulnerability response plans.
   FDA's acceptance now depends on documentation of robust cybersecurity processes in the quality system, not just best-practice recommendations.
3. **Cybersecurity Transparency & Lifecycle Ops**
   Includes Section VII (Cyber Devices) for the first time—laying out pre- and post-market cybersecurity obligations tied to § 524B.
   Requires evidence of processes to monitor, identify, and respond to cybersecurity vulnerabilities throughout the device lifecycle.
   Calls for clear labeling and instructions on cybersecurity aspects, and emphasizes vendor and third-party component management.

Expanding on the three above-mentioned parts, the 2025 FDA guidance also expands the previous Appendix 1 from 2023 (security control categories and associated recommendations), integrating:

- Categories of security controls
- Requirements for source code custodial control
- Acknowledgement of ANSI/AAMI SW96 and NIST/IEC frameworks as implementation references

These recommendations are described and summarized in Table 1.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 1. FDA: security control categories

Security category	Description
Authentication	Two types of authentication should be implemented: the authentication of information and the authentication of entities. The former proves that the information has come from a known and trusted source, the latter proves the identity for an endpoint or authorized user.
Authorization	It refers to the rights or the permissions granted to an entity such as a device or user to access the requested resource.
Cryptography	Recommendations to implement the required algorithms to meet secure-by-design objectives. This recommendation focuses on the selection and implementation of appropriate cryptographic schemes and protocols.
Code, data, and execution integrity	These recommendations tackle cybersecurity concerns related to code, data, and execution integrity. Appropriate cybersecurity measures must be put in place to ensure the integrity of these elements is guaranteed at all times.
Confidentiality	These recommendations focus on mechanisms required to keep data (in transit or at rest) confidential by means of appropriate encryption algorithms.
Event detection and logging	All security events should be properly detected and logged so that they can be retrieved and analyzed in case of security incidents. The integrity and availability of these logs must be ensured to prevent attackers from deleting or tampering with logs.
Resilience and recovery	These recommendations address those requirements that allow a device to be resistant to cyberattacks and eventually recover even after severe fault conditions.
Updatability and patchability	A device must be able to be updated securely when the device is deployed. This is even more important as attacks become more sophisticated as time goes on and old hardware/software becomes less reliable.

3.2 MDR regulation

The MDR is a regulation governing the production and distribution of medical devices in the EU that came into effect in 2017. It aims to ensure the safety, quality and effectiveness of medical devices while enhancing transparency and traceability throughout the supply chain. It replaces the old *Medical Device Directive (MDD)* and introduces stricter requirements for manufacturers of IoMT devices. The MDR includes, among other things, provisions to address cybersecurity concerns related to IoMT devices to ensure that medical devices are designed and manufactured with adequate cybersecurity measures to protect against potential threats. The structure and contents of the MDR is schematized in Figure 3.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

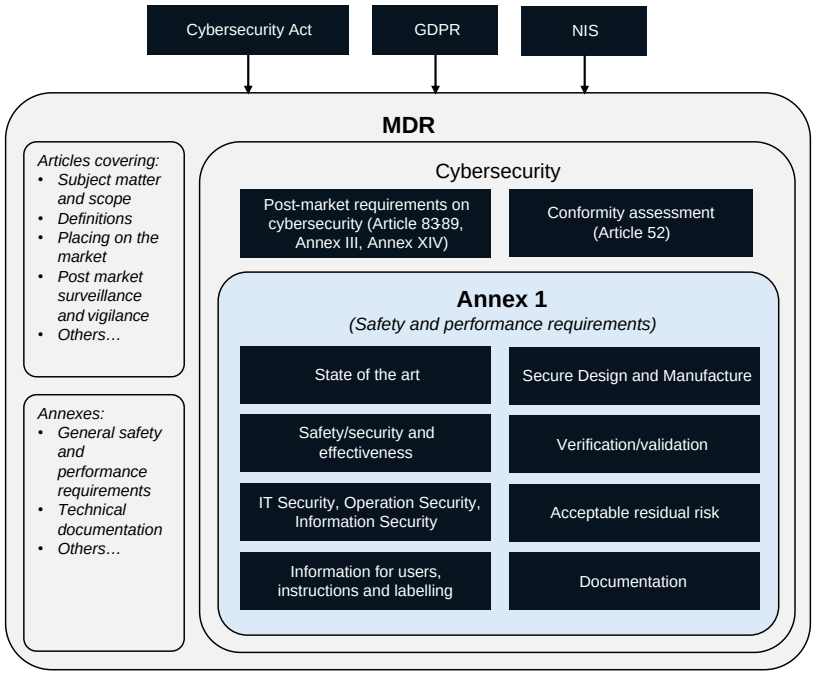

Figure 3. MDR regulation structure

The MDR regulation consists of 123 articles and 12 annexes covering different aspects related to quality, security and transparency of IoMT devices. In particular, cybersecurity provisions are covered in the following articles and sections:

- **Articles 83-89 and Annexes III-XIV (Post-market requirements on cybersecurity)**: these articles describe the post-market requirements that IoMT devices must comply to in order to meet the regulation. These requirements include the systematic gathering, recording, and analysis of relevant data on the quality, performance, and safety of a device throughout its entire lifetime. This data should then be used to detect issues and improve on the product quality;
- **Article 52 (Conformity assessment)**: describes the conformity process that a manufacturer must execute on devices to make sure they comply with the regulation.
- **Annex 1 (Safety and performance requirements)**: describe some broad safety requirements for both pre-market and post-market phases. These requirements include things such as protection against radiations, protection against mechanical, and thermal risks, information on the label, etc. It also outlines cybersecurity requirements that the device must meet to comply with the regulation.

Following the MDR regulation, a guidance document (***MDCG 2019-16***) titled *Guidance on Cybersecurity for medical devices* was released to assist medical device manufacturers on meeting the cybersecurity requirements set out by the MDR. The document goes into extensive detail on how manufacturers and other actors can meet the requirements set out in *Annex 1* of the MDR. In particular, the guidance details eight practices that the manufacturer should employ in order to achieve the requirements set out by the MDR. These eight practices are summarized in Table 2:

Table 2. MDR: security practices

Security category	Description
Security management	This security practice addresses the security-related activities and ensures that they are sufficiently planned, documented, and completed during a product's life cycle.
Specification of security requirements	This security practice deals with identifying the security capabilities needed for the product to achieve sufficient confidentiality, integrity and availability of data, functions, and services.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 2. MDR: security practices*...continued*

Security category	Description
Secure by design	This security practice defines processes to ensure that the product is secure by design.
Secure implementation	Defines processes to ensure that product features are implemented securely. This is relevant for all hardware and software components of the product.
Security verification and validation testing	Covers the definition of the testing activities required to ensure that all security requirements are met. This practice also ensures that the product security is maintained while the product is in use.
Management of security-related issues	Describes processes addressing how security matters will be handled if and when they arise.
Security update management	Describes processes to address how security updates will be tested for regressions and rolled out in a timely manner.
Security guidelines	This practice details how to create and maintain user guidelines for the final product once it is out in the field. The documentation describes, but is not limited to, how to integrate, configure and maintain the security strategy.

Moreover, the *MDCG 2019-16* document outlines in Section 3.3 an indicative list of **security capabilities** that can be used as a basis to comply with MDR *Annex 1* requirements. Among these security capabilities we can find:

- **Configuration of security features**
- **Cybersecurity product upgrades**
- **Data backup and disaster recovery**
- **Personal data integrity and authenticity**
- **Node authentication**
- **Transmission confidentiality**

.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

4 Ease MDR and FDA cyber security requirements with EdgeLock SE05x secure elements and A5000

This section will list the security requirements and recommendation outlines in both the FDA and MDR regulations and how EdgeLock SE05x/A5000 can be used in medical devices for compliance to the security requirements of such regulations. Figure 4 gives on overview on the secure element features which support the FDA and MDR regulations.

Note: *Only security requirements that can be fully or partially met with NXP solutions are listed in this section. For a complete list of requirements, including software and hardware requirements, please refer to the standard/ protocol specification.*

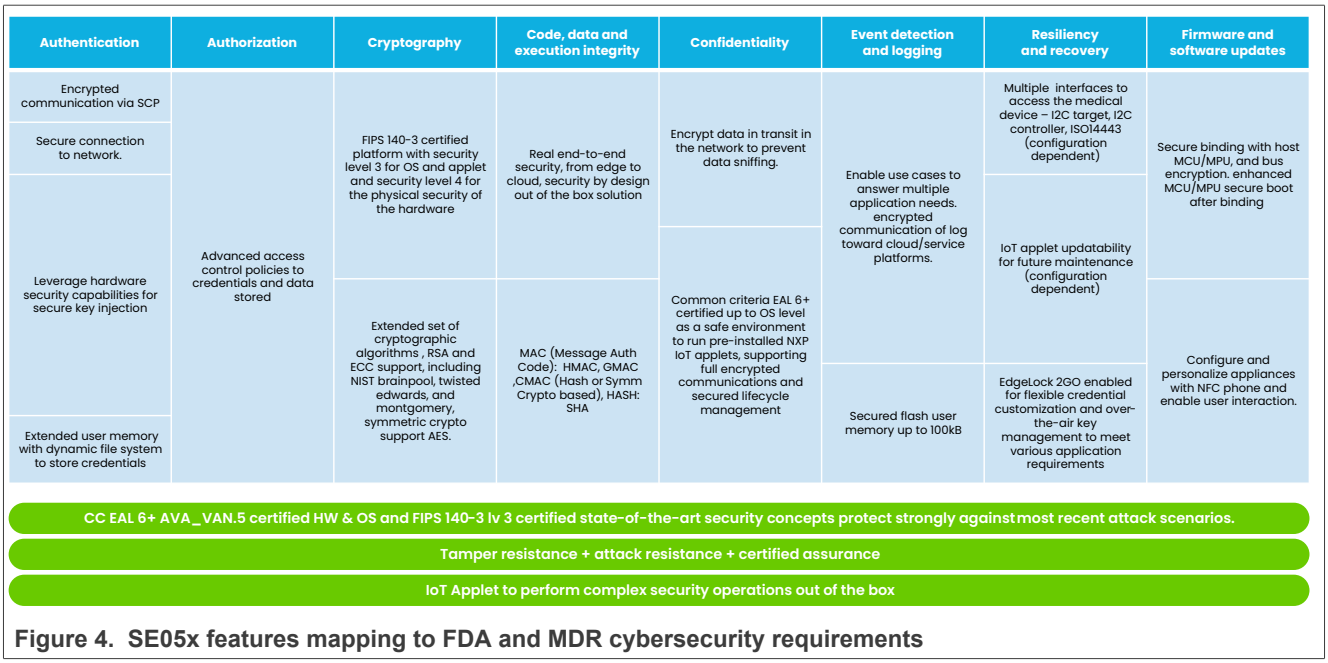

Figure 4. SE05x features mapping to FDA and MDR cybersecurity requirements

4.1 Meeting FDA requirements with EdgeLock SE05x/A5000

This section focuses on how EdgeLock SE05x/A5000 can be used to facilitate compliance with the security controls outlined in *Appendix 1* of the FDA guidance.

The security requirements for the FDA are divided into the following categories:

- **Authentication**
- **Authorization**
- **Cryptography**
- **Code, Data, and Execution Integrity**
- **Confidentiality**
- **Event Detection and Logging**
- **Resiliency and Recovery**
- **Firmware and Software Updates**

The FDA goes through each requirement in the pre-submission document in great detail and provides considerations for each category. For each submission, the submitter has to explain how their products meet the requirements and in case of waiving a requirement, what is the justification for the waiving. This application note will explain how to meet these outlined requirements by integrating NXP solutions.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

The following sections walk the FDA requirements shown in Figure 5 in detail.

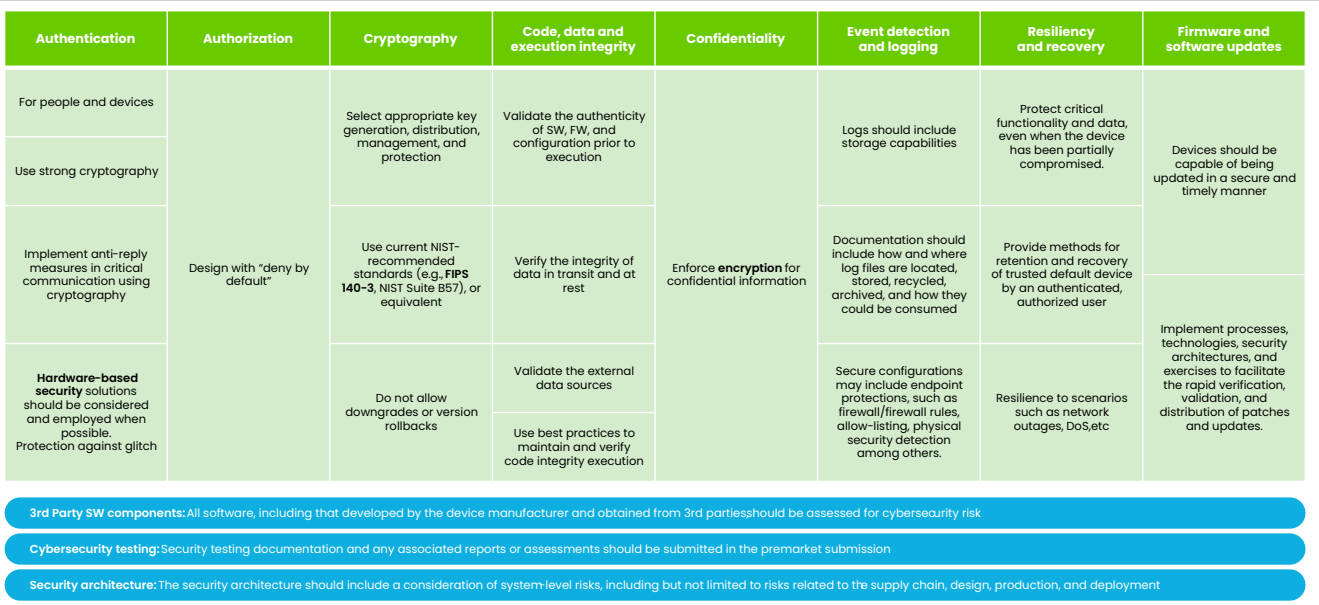

Figure 5. FDA: cybersecurity requirements overview

4.1.1 Authentication

The authentication is split up into two separate controls for the FDA. There is authentication of information and authentication of entities. The medical device needs to be able to prove the authenticity of data produced as well as verify the authenticity of data received from external sources. Entities such as user or nodes in the system also need to be authenticated. The data that needs to be authenticated is as follows:

1. Information at rest (stored).
2. Information in transit (transmitted).
3. Entity authentication of communication endpoints such as device-to-device authentication.
4. Software binaries.
5. Integrity of the execution state of currently running software.
6. Any other appropriate part of the medical device system identified in the devices threat model.

Meet 1, 2, 3, 4, 5, 6: A secure authentication should be implemented, such as a Transport Layer Security (TLS) handshake. A TLS handshake is often used to establish a secure and authenticated connection between two nodes over the Internet. EdgeLock A30 supports the cryptographic authentication requirements and operations defined by TLS v1.2/v1.3 using ECC keys. Both the ECDH(E) algorithm for key agreement and the ECDSA algorithm for digital signatures are supported. The ECC curves supported include NIST P-256 and Brainpool P256r1. AES symmetric keys of 128 and 256 bits are supported and can be used in ECB, CBC, CTR, GCM, and CCM operation modes for data encryption. Finally, the SHA-256 and SHA-384 algorithms are supported as well. To simplify the integration of TLS, the NX middleware provides an mbedTLS ALT implementation that allows mbedTLS stack to use the secure authenticator to perform the authentication crypto operations that are part of the TLS handshake between client and server. Such a plugin is as well available for OpenSSL and PKCS#11. Within the A30 keys for both asymmetric and symmetric cryptography can be stored. There is a secure user memory available that can be used to store credentials. The following NXP material can be used as a starting point to meet the requirements listed above:

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 3. NXP material: Authentication

NXP material	Relevant content
NX middleware APIs	• **Read certificate / binary data / certificate:** sss_key_store_get_key (), nx_ReadData() • **Inject a certificate / binary data / certificate:** sss_key_store_set_key (), nx_WriteData() • **Key creation:** sss_key_store_generate_key () • **Key import / injection:** sss_key_store_set_key(), nx_ManageKeyPair(), nx_ChangeKey() • **ECDH Key Derivation:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2(), sss_derive_key_one_go() • **Read EdgeLock A30 pre-injected UID:** nx_GetCardUID()
NX middleware demos and examples	• **OpenSSL Engine/Provider: TLS Client example** • **AWS Demo for Raspberry Pi** • **AWS Cloud Demo on FreeRTOS** • **MCXN-947 AWS Cloud Demo** • **Symmetric AES Encryption example (key injection):** ex_symmetric • **ECC Signing/Verifing example (key generation):** ex_ecc • **ECDH Key Derivation example** ex_ecdh • **Get UID example:** ex_get_uid
Application notes	• AN14238 Get started with EdgeLock A30 support package • AN14559 Migration Guide from EdgeLock A5000 to EdgeLock A30

4.1.2 Authorization

Authorization is the set of permissions required for a system resource to be used by an entity such as a device. For example, this could be a programmer attempting to reprogram a pacemaker or a nurse attempting to access patient data. A well defined and implemented authorization scheme would assign entities the minimum required level for the entity to perform all tasks. The recommendations from the FDA are as follows:

1. Limit authorized access to devices through the authentication of users.

Meet 1: EdgeLock SE05x/A5000 can employ the use of UIDs to enable authorizations; each UID would have a certain level of access. These UIDs would be attributed to each user or entity such as another device. When establishing a connection, the entity would need to sign the UID. This can then be verified by the EdgeLock SE05x/A5000. The following NXP material can be used as a starting point to meet the requirements listed above:

Table 4. NXP material: Authorization functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **Read EdgeLock SE05x pre-injected UID**: Se05x_API_ReadObject() • **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()
Plug & Trust middleware demos and examples	• **Get Info example (retrieve SE UID)**: \simw-top\demos\se05x\se05x_GetInfo • **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa

4.1.3 Cryptography

Cryptographic algorithms and protocols are recommended to meet the secure by design objectives. There are already existing standardized cryptographic algorithms. The FDA recommendations when selecting and implementing the use of cryptography are as follows:

1. Select an industry standard cryptography scheme.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

2. Use current NIST recommended standards for cryptography such as FIPS 140-3, NIST Suite B57, or equivalent.

Meet 1, 2: EdgeLock SE05x/A5000 allows for many forms of encryption to be used. Both symmetric and asymmetric cryptographic algorithms are supported by the EdgeLock SE05x/A5000. An extensive set of cryptographic functions are supported by the EdgeLock SE05x/A5000 including AES, 3DES, RSA, ECC. The ECC curves supported include NIST (up to 521 bits key length), Brainpool, Twisted Edwards and Montgomery. MACing (HMAC, CMAC, GMAC), hashing (SHA-1, SHA-224/256/384/512), TRNG compliant to NIST SP800-90B, and DRBG compliant to NIST SP800-90A. These encryption methods are industry used and recommended standards and NIST recommended. Not only the SE050F is FIPS 140-2 certified, but the latest addition to the EdgeLock family SE052F is FIPS 140-3 level 3 certified with level 4 for physical security. Listed below are the relevant NXP materials that can assist in the implementation of the requirements specified above:

Table 5. NXP material: Cryptographic functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **Symmetric encryption and decryption operations:** sss_cipher_crypt_ctr (), sss_cipher_one_go (), sss_cipher_one_go_v2 () • **Asymmetric encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go() • **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot() • **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()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric • **HMAC Example:** simw-top\sss\ex\hmac • **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa

4.1.4 Code, data, and execution integrity

The integrity of a device is critical as many cybersecurity attracts target the devices integrity in some form or other. This could be on the stored code, the date, or even the execution state. The FDA recommendations are split into the 3 categories and are as follows:

1. **Code Integrity**
   a. Hardware-based security solutions should be employed when possible.
   b. Authenticate firmware and software.
   c. Allow installation of cryptographically authenticated firmware and software updates.
   d. Ensure that the authenticity of software is validated before being executed.
   e. Disable/restrict unauthorized access to test and debug ports.
   f. Use tamper-evident seals on device enclosure and ports to verify physical integrity.
2. **Data Integrity**
   a. Verify the integrity of all incoming data.
   b. Validate that all data originating from external sources is compliant with the protocol.
   c. Protect the integrity of data that is necessary for the device safety.
3. **Execution Integrity**
   a. Use best practices from industry to maintain and verify the integrity of the code while it is being executed.

It is important to note that the integrity is separated into 2 distinct sections. There is the integrity of the: code, data, and execution within the EdgeLock SE05x/A5000. This has been ensured with the CC 6+ certification. The

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

second part is the integrity of the host. In this second part, the EdgeLock SE05x/A5000 would be used with the host to ensure the integrity.

Meet 1.a: EdgeLock SE05x/A5000 is a hardware solution that is easy to implement into a system during design. EdgeLock SE05x/A5000 is pre-provisioned with keys and credentials in a highly secure and controlled environment.

Meet 1.b, 1.c, 1.d: EdgeLock SE05x/A5000 can securely store the ECC private key or the RSA private key (only certain SE05x models) that is required to decrypt a firmware update that has been encrypted using the associated public key. The private key is protected in the common criteria certified SE tamper-resistant hardware and never leaves the boundaries of the SE secure environment. Symmetric encryption / decryption is supported as well using AES (ECB, CBC, CTR, GCM, CCM) and 3DES. EdgeLock SE05x/A5000 can securely store the public key required to verify the firmware signature. For verifying that the signature, ECDSA and SHA algorithms are supported (additionally EdgeLock SE05x supports EdDSA and RSA).

Meet 2.a, 2.b, 2.c, 3.a: It is advised to avoid using only CRC checks to ensure the integrity of data as it is not secure. EdgeLock SE05x/A5000 can handle the integrity of code/data one of two ways. The first is with the use of use of a MAC such as: HMAC, GMAC, or CMAC. The second is hashing with the following algorithms: SHA-1, SHA-224, SHA 256, SHA-384, SHA-512.

The following NXP material can be used as a starting point to meet the requirements listed above:

Table 6. NXP material: Integrity functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot() • **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()
Plug & Trust middleware demos and examples	• **HMAC Example:** simw-top\sss\ex\hmac • **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa
Application notes	• AN13013 Get started with EdgeLock SE05x support package • AN12450 EdgeLock SE05x Quick start guide with i.MX RT1060 and i.MX RT1170 • AN12663 EdgeLock® SE05x to implement TPM-like functionality • AN12449 Sensor data protection with EdgeLock SE05X

4.1.5 Confidentiality

Support should be in place to allow the confidentiality of all data which could lead to patient harm. If the authentication and authorization of the system has been implemented correctly, confidentiality is mostly assured. The EdgeLock SE05x/A5000 can encrypt data before it is transmitted into the network to safeguard it from being intercepted. The EdgeLock SE05x/A5000 is Common Criteria EAL 6+ certified up to OS level to run the pre-installed NXP IoT applets. Refer to the guidance laid out in Section 4.1.1 and Section 4.1.2 for guidance on implementing confidentiality.

Support should be in place to allow the confidentiality of all data which could lead to patient harm. If the authentication and authorization of the system has been implanted correctly, confidentiality is mostly assured. The EdgeLock SE05x/A5000 can encrypt data before it is transmitted into the network to safeguard it from being intercepted. Refer to the guidance laid out in Section Section 4.1.1, Section 4.1.2, and Section 4.1.3 for guidance on implementing confidentiality.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

4.1.6 Event detection and logging

In the case of an attack, it is vital for the system to have measures in place for detecting the attack and logging the necessary data. To assist with detecting an attack, all the previously mentioned authentication and integrity checks can be used in identify if there is an attempted attack. Once an attack has been detected the system must keep a log for it to be investigated at a later time. The FDA requirement is as follows:

1. The device should be able to securely create and store log files off the device to track attempted attacks.

Event detection and logging is more linked to being a system function. However, there are ways in which the EdgeLock SE05x/A5000 can be used in conjunction with the host to ensure the data can be logged securely.

Meet 1: Once an attack is detected EdgeLock SE05x/A5000 can log and be used to securely log data into the secure memory, up to 100 kB of user memory in the case of the SE052. To ensure further security of the data, EdgeLock SE05x/A5000 can ensure the confidentiality and integrity of the data by encrypting, MACing/hashing the data. This can then allow the device to send data via an unsecure network to keep data off the device. The Global Platform Secure Channel Protocol 03 (SCP03) is natively supported by EdgeLock SE05x, details on this can be found in AN12662. SCP03 can be used to ensure the connection between host and the EdgeLock SE05x/A5000 is authentic, confidential, and has integrity.

Table 7. NXP material: Event detection and logging functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **Symmetric encryption and decryption operations:** sss_cipher_crypt_ctr (), sss_cipher_one_go (), sss_cipher_one_go_v2 () • **Asymmetric encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go() • **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric • **HMAC Example:** simw-top\sss\ex\hmac

4.1.7 Resiliency and recovery

Devices can often come under attack. It is critical, especially in healthcare applications, that these devices continue to work as intended. Devices need to be able to maintain function while under attack. The recommendation to achieve this “cyber-resilience” is as follows:

1. Design devices to provide methods for retention and recovery of trusted default devices configurations by an authenticated, authorized user.

Meet 1: Once the EdgeLock SE05x/A5000 has been set up, and the keys injected all crypto algorithms can still be performed once a connection has been established and communication has not been interrupted. The physical connection to the SE will be via the I2C interface. This can ensure that even if taken offline the device can still authenticate users. If the communication with the cloud has been severed EdgeLock SE05x/A5000 would be able to securely store data in the secure storage.

Listed below is material from NXP that can help meet the requirement set above:

Table 8. NXP material: Event detection and logging functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **Symmetric encryption and decryption operations:** sss_cipher_crypt_ctr (), sss_cipher_one_go (), sss_cipher_one_go_v2 () • **Asymmetric encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go()

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 8. NXP material: Event detection and logging functions...*continued*

NXP material	Relevant content
	• **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric • **HMAC Example:** simw-top\sss\ex\hmac

4.1.8 Firmware and software updates

Devices should be able to be updated securely and quickly. Beyond this, the FDA also recommends manufacturers also plan for rapid testing, evaluation, and patching of devices once in the field. The recommendations are as follows:

1. Design devices to allow firmware and software patches.
2. Implement a secure process for providing validated software updates and patches for users.

As previously mentioned in Section 4.1.4, there are two sections to the system. The EdgeLock SE05x/A5000 and the host. The EdgeLock SE05x/A5000 being CC 6+ compliant can ensure with its integrity protection that only valid updates will be accepted. However, the host must use the EdgeLock SE05x/A5000 in order to ensure the updates are compliant with the standard.

Meet 1: The EdgeLock SE051 integrates SEMS Lite technology to allow you to update the NXP IoT Applet. This allows you to have the latest security updates from NXP and stay up to date with the ever-changing security requirements as attacks become more sophisticated. Certain variations of the SE allow you to upload custom applets.

Meet 2: EdgeLock SE05x/A5000 can securely store the ECC private key or the RSA private key (EdgeLock SE05x only) that is required to decrypt a firmware update that has been encrypted using the associated public key. The private key is protected in the SE tamper-resistant hardware and never leaves the boundaries of the SE secure environment. Symmetric encryption / decryption is supported as well using AES (ECB, CBC, CTR, GCM, CCM) and 3DES. EdgeLock SE05x/A5000 can securely store the public key required to verify the firmware signature. The public key can be protected from deletion and overwriting (or other unintended usage) using secure object policies. For verifying the signature, ECDSA and SHA algorithms are supported (additionally EdgeLock SE05x supports EdDSA and RSA).

Table 9. NXP material: Secure firmware update

NXP material	Relevant content
Plug & Trust middleware APIs	• **Encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go() • **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot() • **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()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric • **Message Digest Example:** \simw-top\sss\ex\md • **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa
Application notes	• AN12907 Secure update of EdgeLock™ SE051 IoT applet

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 9. NXP material: Secure firmware update...*continued*

NXP material	Relevant content
	• AN12909 How to develop JCPO applets on EdgeLock SE051 using JCOP Tools (SE051 secure files)
Training	• Keeping the Security of Your Devices Up To Date with the EdgeLock SE051 Secure Element and EdgeLock® 2GO Service • Secure Element Common | SEMS Tooling | Part 2: SEMS Lite Tooling

4.2 Meeting MDR requirements with EdgeLock SE05x/A5000

Within Section 3.3 of the MDCG 2019/16, an indicative list of security capabilities has been shared to ensure the device security capabilities. The share list should be used to help guide the design process of medical devices. A more concise list:

- Authentication
- Authorization
- Confidentiality
- Integrity
- Cybersecurity Product Upgrades
- Configuration of Security Features

This section focuses on how EdgeLock SE05x/A5000 can be used to facilitate compliance with some of the security capabilities outlined in *MDCG 2019-16*

4.2.1 Authentication

Authentication is referred to in three separate requirements listed in the MDCG 2019/16:

1. The authenticity of a node.
2. Personal data authenticity.
3. Person authentication.

For 1, 2, 3: EdgeLock SE05x/A5000 supports the cryptographic requirements and operations defined by TLS v1.2/v1.3 using both ECC and RSA keys. For ECC keys, both the ECDHE algorithm for key agreement and the ECDSA algorithm for digital signatures are supported. AES symmetric keys of up to 256 bits are supported and can be used in ECB, CBC, CTR, GCM, and CCM operation modes for data encryption. Hashing algorithms such as SHA-256 and SHA-384 are supported as well. To simplify the integration of TLS, the Plug & Trust middleware an implementation, which allows mbedTLS stack to use the secure element to perform the crypto operations that are part of the TLS handshake between client and server. Such a plugin is as well available for OpenSSL and PKCS#11. Within the EdgeLock SE05x/A5000 keys for both asymmetric and symmetric can be stored. There is a secure user memory available that can be used to store credentials. The following NXP material can be used as a starting point to meet the requirements listed above:

Table 10. NXP material: Authentication

NXP material	Relevant content
Plug & Trust middleware APIs	• **Perform TLS handshake:** Se05x_API_TLSGenerateRandom (), for TLS v1.2: Se05x_API_TLSCalculatePreMasterSecret () , Se05x_API_TLSPerformPRF() • **Get handle of (pre)provisioned keys or objects:** sss_se05x_key_object_get_handle(), sss_key_store_get_key () • **Key creation:** sss_se05x_key_store_generate_key () • **Key import / injection:** sss_key_store_set_key(), Se05x_API_WriteECKey (), Se05x_API_WriteRSAKey ()

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 10. NXP material: Authentication*...continued*

NXP material	Relevant content
	• **Read EdgeLock SE05x pre-injected UID:** Se05x_API_ReadObject(kSE05x_AppletResID_ UNIQUE_ID)
Plug & Trust middleware demos and examples	• **TLS Client example**: \simw-top\demos\linux\tls_client • **Inject Certificate into SE example**: \simw-top\demos\se05x\se05x_InjectCertificate • **Get Certificate from the SE**: \simw-top\demos\se05x\se05x_GetCertificate • **Get Info example** (retrieve SE UID): \simw-top\demos\se05x\se05x_GetInfo • **Using policies for secure objects demo:** \simw-top\demos\se05x\se05x_policy • **EdgeLock 2GO Agent examples**:\simw-top\nxp_iot_agent\ex
Application notes	• AN12399 EdgeLock SE05x for device-to-device authentication • AN12400 EdgeLock SE05x for secure connection to OEM cloud

4.2.2 Authorization

Authorization is defined as the access rights entities within the system have to other parts of the system. This could be a device in the system needing to be authorized to receive data from the server. A well defined and implemented authorization scheme would assign entities the minimum required level for the entity to perform all tasks. Examples of entities that need authorization are as follows:

1. Authorization of users.
2. Authorization of devices.

Meet 1, 2: EdgeLock SE05x/A5000 can employ the use of UIDs to enable authorizations, each UID would have a certain level of access. These UID would be attributed to each user or entity such as another device. Within the backend of the system the access rights can be stored. When establishing a connect, the entity would need to communicate its UID and provide some sort of integrity such as signing. This can then be verified by the EdgeLock SE05x/A5000. The following NXP material can be used as a starting point to meet the requirements listed above:

Table 11. NXP material: Authorization

NXP material	Relevant content
Plug & Trust middleware APIs	• **Read EdgeLock SE05x pre-injected UID**: Se05x_API_ReadObject() • **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()
Plug & Trust middleware demos and examples	• **Get Info example** (retrieve SE UID): \simw-top\demos\se05x\se05x_GetInfo • **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa

4.2.3 Confidentiality

It is vital to ensure that information within the system, whether it is stored or transmitted, is confidential. Outlined within the MDCG 2019/16 are three capabilities specifically for confidentiality. They are as follows:

1. Personal data de-identification
2. Personal data storage confidentiality
3. Transmission confidentiality

Meet 1,2,3: EdgeLock SE05x/A5000 supports the cryptographic requirements using both ECC and RSA keys. EdgeLock SE05x/A5000 securely stores the private keys. The ECC curves supported include NIST (up to 521 bits key length), Brainpool, Twisted Edwards and Montgomery. MACing (HMAC, CMAC, GMAC), hashing (SHA-1, SHA-224/256/384/512), TRNG compliant to NIST SP800-90B, and DRBG compliant to NIST

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

SP800-90A. Symmetric encryption and decryption is also supported using AES (ECB, CBC, CTR, GCM, CCM) and 3DES. AES symmetric keys of up to 256 bits are supported and can be used in ECB, CBC, CTR, GCM and CCM operation modes for data encryption. The following NXP material can assist in meeting the confidentiality requirements:

Table 12. NXP material: Confidentiality

NXP material	Relevant content
Plug & Trust middleware APIs	• **Key creation:** sss_se05x_key_store_generate_key () • **Key import / injection:** sss_key_store_set_key(), Se05x_API_WriteECKey (), Se05x_API_Write RSAKey () • **Asymmetric encryption and decryption operations:** sss_asymmetric_encrypt(), sss_ asymmetric_decrypt(), sss_cipher_one_go() • **Symmetric encryption and decryption operations:** sss_cipher_crypt_ctr (), sss_cipher_one_ go (), sss_cipher_one_go_v2 ()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric

4.2.4 Integrity

The methods to ensure the integrity of data change depending on the cryptographic method chosen. The importance of ensuring integrity is to ensure the data arriving to the device has not been modified and is the correct data. The two capabilities highlighted in the MDCG/2019/16 are:

1. Personal data integrity
2. Transmission integrity

It is important to note that the integrity is separated into 2 separate sections. There is the integrity of the EdgeLock SE05x/A5000. This has been ensured with the CC 6+ certification. The second part is the integrity of the host. In this second part the EdgeLock SE05x/A5000 would be used with the host to ensure the integrity.

Meet 1, 2: EdgeLock SE05x/A5000 can handle the integrity of code/data one of two ways. The chosen cryptography method determines the required operation for integrity. The first is the with the use of a MAC such as: HMAC, GMAC, or CMAM. The second is hashing with the following algorithms: SHA-1, SHA-224, SHA 256, SHA-384, SHA-512.

Table 13. NXP material: Integrity functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_ mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot() • **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()
Plug & Trust middleware demos and examples	• **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa
Application notes	• AN13013 Get started with EdgeLock SE05x support package • AN12450 EdgeLock SE05x Quick start guide with i.MX RT1060 and i.MX RT1170 • AN12663 EdgeLock® SE05x to implement TPM-like functionality • AN12449 Sensor data protection with EdgeLock SE05X

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

4.2.5 Cybersecurity product upgrades

If you integrate the **EdgeLock SE051** or **SE052** SE in your IoT solution, you can take advantage of the **SEMS Lite technology** to update the SE on-the-field, both online or offline, to always get the latest security patches from NXP and the latest updates required to keep up with the specification as it evolves over time. With EdgeLock SE051, devices can take advantage of the latest features and security improvements as soon as they are available and always enjoy a high protection level for stored credentials.

Table 14. NXP material: Cybersecurity Product upgrades

NXP material	Relevant content
Plug & Trust middleware APIs	• **Encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go() • **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot() • **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()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption Example:** \simw-top\sss\ex\symmetric • **ECC Signing Example:** \simw-top\sss\ex\ecc • **RSA Signing Example:** \simw-top\sss\ex\rsa
Application notes	• AN12907 Secure update of EdgeLock™ SE051 IoT applet
Training	• Keeping the Security of Your Devices Up To Date with the EdgeLock SE051 Secure Element and EdgeLock® 2GO Service • Secure Element Common | SEMS Tooling | Part 2: SEMS Lite Tooling

4.2.6 Configuration of security features

It is vital while a system is being developed that the security configurations such as keys used or authorization permissions can be molded to fit the use case better. EdgeLock SE05x/A5000 can enable this flexibility as the developer is able to customize keys and credentials beyond the scope of the Ease-of-Use configuration. This can be done through the process of adding secure objects, creating crypto object, a factory reset, and more. There is also the option to configure the EdgeLock SE05x/A5000 once it has been deployed through the contactless interface. Listed below are NXP material recommendations to assist:

Table 15. NXP material: Configuration of security features

NXP material	Relevant content
Plug & Trust middleware APIs	• **Encryption and decryption operations:** sss_asymmetric_encrypt(), sss_asymmetric_decrypt(), sss_cipher_one_go() • **MACing operation:** sss_mac_context_free (), sss_mac_context_init (),sss_mac_finish(),sss_mac_init(),sss_mac_one_go(),sss_mac_update() • **Hashing operations:** sss_se05x_digest_one_go (), Se05x_API_DigestOneShot() • **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()
Plug & Trust middleware demos and examples	• **Personalization of SE051:** DEMO for Personalization of SE051
Application notes	• AN12436 SE050 configurations

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5 Ease MDR and FDA cyber security requirements with EdgeLock A30

This section will list the security requirements and recommendation outlines in both the FDA and MDR regulations and how EdgeLock A30 can be used in medical devices for compliance to the security requirements of such regulations. Figure 6 gives on overview on the secure element features which support the FDA and MDR regulations.

Note: *Only security requirements that can be fully or partially met with NXP solutions are listed in this section. For a complete list of requirements, including software and hardware requirements, please refer to the standard/ protocol specification.*

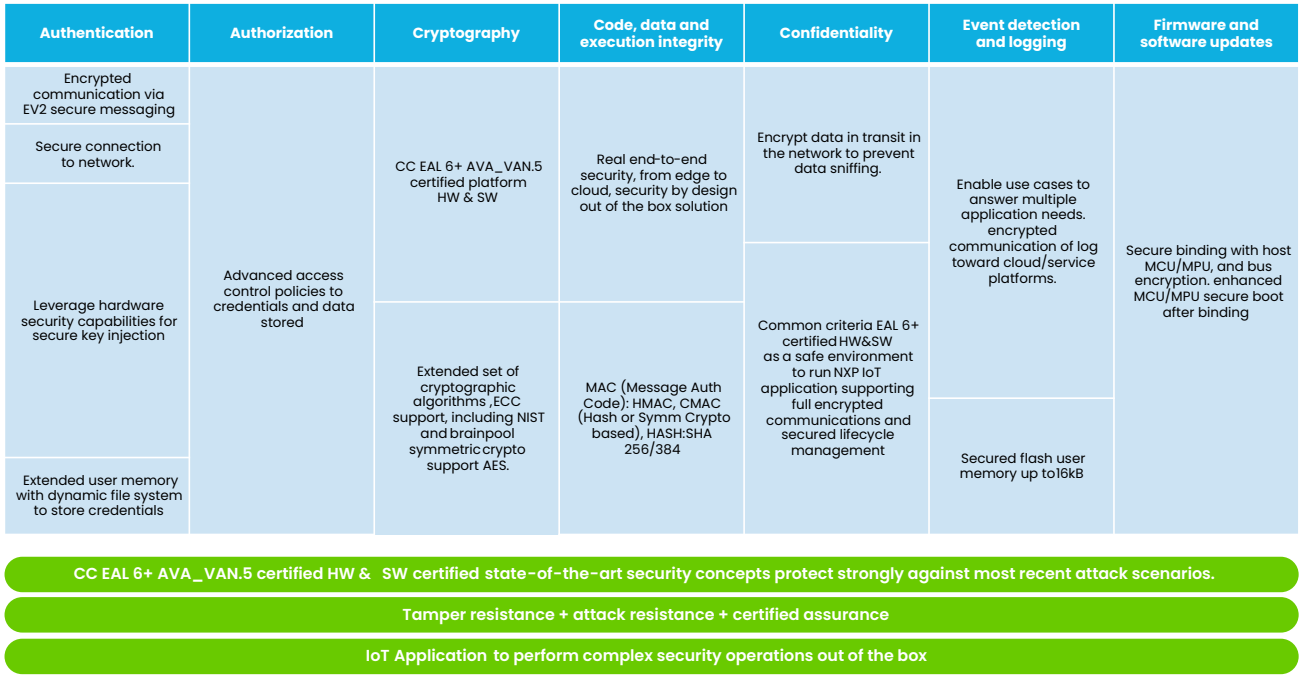

Figure 6. EdgeLock A30 features mapping to FDA and MDR cybersecurity requirements

5.1 Meeting FDA requirements with EdgeLock A30

This section focuses on how EdgeLock A30 can be used to facilitate compliance with the security controls outlined in *Appendix 1* of the FDA guidance.

The security requirements for the FDA are divided into the following categories:

- **Authentication**
- **Authorization**
- **Cryptography**
- **Code, Data, and Execution Integrity**
- **Confidentiality**
- **Event Detection and Logging**
- **Resiliency and Recovery**
- **Firmware and Software Updates**

The FDA goes through each requirement in the pre-submission document in great detail and provides considerations for each category. For each submission, the submitter has to explain how their products meet

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

the requirements and in case of waiving a requirement, what is the justification for the waiving. This application note will explain how to meet these outlined requirements by integrating NXP solutions.

The following sections walk the FDA requirements shown in Figure 7 in detail.

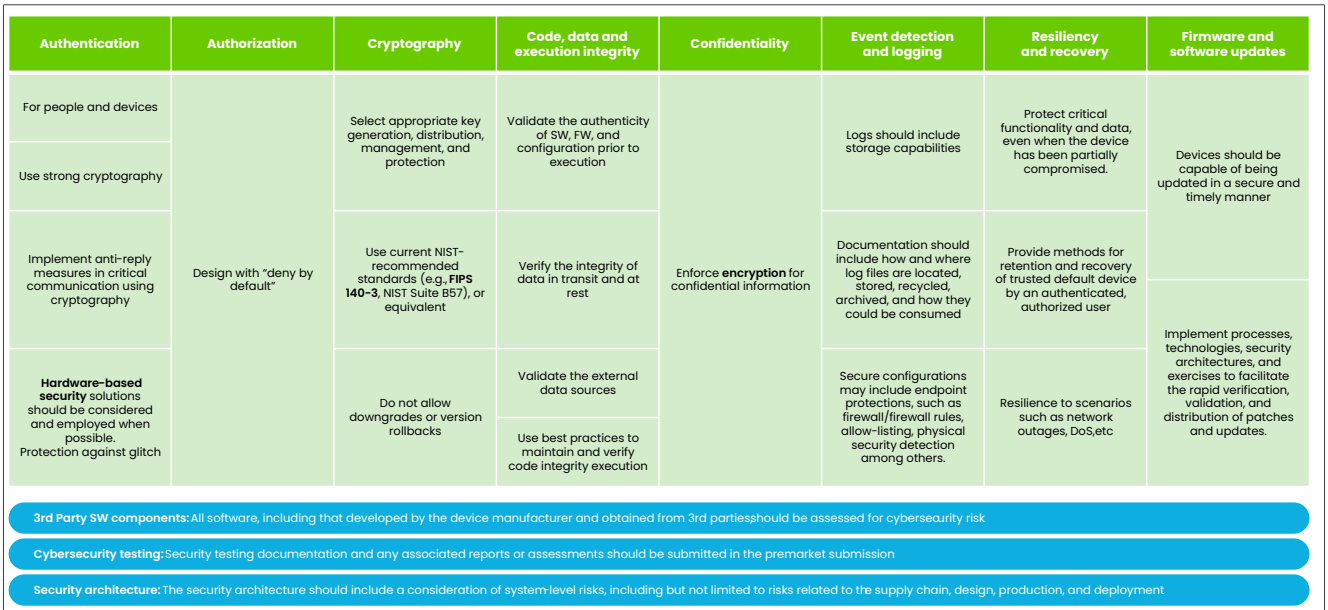

Figure 7. FDA: cybersecurity requirements overview

5.1.1 Authentication

The authentication is split up into two separate controls for the FDA. There is authentication of information and authentication of entities. The medical device needs to be able to prove the authenticity of data produced as well as verify the authenticity of data received from external sources. Entities such as user or nodes in the system also need to be authenticated. The data that needs to be authenticated is as follows:

1. Information at rest (stored).
2. Information in transit (transmitted).
3. Entity authentication of communication endpoints such as device-to-device authentication.
4. Software binaries.
5. Integrity of the execution state of currently running software.
6. Any other appropriate part of the medical device system identified in the devices threat model.

Meet 1, 2, 3, 4, 5, 6: A secure authentication should be implemented, such as a Transport Layer Security (TLS) handshake. A TLS handshake is often used to establish a secure and authenticated connection between two nodes over the Internet. EdgeLock A30 supports the cryptographic authentication requirements and operations defined by TLS v1.2/v1.3 using ECC keys. Both the ECDH(E) algorithm for key agreement and the ECDSA algorithm for digital signatures are supported. The ECC curves supported include NIST P-256 and Brainpool P256r1. AES symmetric keys of 128 and 256 bits are supported and can be used in ECB, CBC, CTR, GCM, and CCM operation modes for data encryption. Finally, the SHA-256 and SHA-384 algorithms are supported as well. To simplify the integration of TLS, the NX middleware provides an mbedTLS ALT implementation that allows mbedTLS stack to use the secure authenticator to perform the authentication crypto operations that are part of the TLS handshake between client and server. Such a plugin is as well available for OpenSSL and PKCS#11. Within the A30 keys for both asymmetric and symmetric cryptography can be stored. There is a secure user memory available that can be used to store credentials. The following NXP material can be used as a starting point to meet the requirements listed above:

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 16. NXP material: Authentication

NXP material	Relevant content
NX middleware APIs	• **Read certificate / binary data / certificate:** sss_key_store_get_key (), nx_ReadData() • **Inject a certificate / binary data / certificate:** sss_key_store_set_key (), nx_WriteData() • **Key creation:** sss_key_store_generate_key () • **Key import / injection:** sss_key_store_set_key(), nx_ManageKeyPair(), nx_ChangeKey() • **ECDH Key Derivation:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2(), sss_derive_key_one_go() • **Read EdgeLock A30 pre-injected UID:** nx_GetCardUID()
NX middleware demos and examples	• **OpenSSL Engine/Provider: TLS Client example** • **AWS Demo for Raspberry Pi** • **AWS Cloud Demo on FreeRTOS** • **MCXN-947 AWS Cloud Demo** • **Symmetric AES Encryption example (key injection):** ex_symmetric • **ECC Signing/Verifing example (key generation):** ex_ecc • **ECDH Key Derivation example** ex_ecdh • **Get UID example:** ex_get_uid
Application notes	• AN14238 Get started with EdgeLock A30 support package • AN14559 Migration Guide from EdgeLock A5000 to EdgeLock A30

5.1.2 Authorization

Authorization is the set of permissions required for a system resource to be used by an entity such as a device. For example, this could be a programmer attempting to reprogram a pacemaker or a nurse attempting to access patient data. A well defined and implemented authorization scheme would assign entities the minimum required level for the entity to perform all tasks. The recommendations from the FDA are as follows:

1. Limit authorized access to devices through the authentication of users.

Meet 1: EdgeLock A30 can employ the use of UIDs to enable authorizations; each UID would have a certain level of access. These UIDs would be attributed to each user or entity such as another device. When establishing a connection, the entity would need to sign the UID. This can then be verified by the EdgeLock A30. The following NXP material can be used as a starting point to meet the requirements listed above:

Table 17. NXP material: Authorization functions

NXP material	Relevant content
NX middleware APIs	• **Read EdgeLock A30 pre-injected UID:** nx_GetCardUID() • **Sign / 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()
NX middleware demos and examples	• **Get UID example:** ex_get_uid • **ECC Signing/Verifing example:** ex_ecc

5.1.3 Cryptography

Cryptographic algorithms and protocols are recommended to meet the secure by design objectives. There are already existing standardized cryptographic algorithms. The FDA recommendations when selecting and implementing the use of cryptography are as follows:

1. Select an industry standard cryptography scheme.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

2. Use current NIST recommended standards for cryptography such as FIPS 140-3, NIST Suite B57, or equivalent.

Meet 1, 2: EdgeLock A30 allows for many forms of encryption to be used. Both symmetric and asymmetric cryptographic algorithms are supported by the EdgeLock A30. An extensive set of cryptographic functions are supported by the EdgeLock A30 including AES and ECC. The ECC curves supported include NIST P-256 and Brainpool P256r1. MACing (HMAC, CMAC), hashing (SHA-256 and SHA-384), TRNG compliant to NIST SP800-90B, and AIS31 class PTG.2. These encryption methods are industry used and recommended standards and NIST recommended. Listed below are the relevant NXP materials that can assist in the implementation of the requirements specified above:

Table 18. NXP material: Cryptographic functions

NXP material	Relevant content
NX middleware APIs	• **Generate a random number:** sss_rng_get_random(), nx_CryptoRequest_RNG() **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish() • **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_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()
NX middleware demos and examples	• **RNG example:** ex_rng • **Symmetric AES Encryption example (key injection):** ex_symmetric • **HMAC Example:** ex_hmac • **Message Digest example:** ex_md • **ECC Signing/Verifing example (key generation):** ex_ecc

5.1.4 Code, data, and execution integrity

The integrity of a device is critical as many cybersecurity attracts target the devices integrity in some form or other. This could be on the stored code, the date, or even the execution state. The FDA recommendations are split into the 3 categories and are as follows:

1. **Code Integrity**
   a. Hardware-based security solutions should be employed when possible.
   b. Authenticate firmware and software.
   c. Allow installation of cryptographically authenticated firmware and software updates.
   d. Ensure that the authenticity of software is validated before being executed.
   e. Disable/restrict unauthorized access to test and debug ports.
   f. Use tamper-evident seals on device enclosure and ports to verify physical integrity.
2. **Data Integrity**
   a. Verify the integrity of all incoming data.
   b. Validate that all data originating from external sources is compliant with the protocol.
   c. Protect the integrity of data that is necessary for the device safety.
3. **Execution Integrity**
   a. Use best practices from industry to maintain and verify the integrity of the code while it is being executed.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

It is important to note that the integrity is separated into 2 distinct sections. There is the integrity of the: code, data, and execution within the A30. This has been ensured with the CC 6+ certification. The second part is the integrity of the host. In this second part, the EdgeLock A30 would be used with the host to ensure the integrity.

Meet 1.a: EdgeLock A30 is a hardware solution that is easy to implement into a system during design. EdgeLock A30 is pre-provisioned with keys and credentials in a highly secure and controlled environment.

Meet 1.b, 1.c, 1.d: EdgeLock A30 can securely store the ECC private key that is required to decrypt a firmware update that has been encrypted using the associated public key. The private key is protected in the common criteria certified SA tamper-resistant hardware and never leaves the boundaries of the SA secure environment. Symmetric encryption / decryption is supported as well using AES-128/256 (ECB, CBC, CTR, GCM, and CCM). EdgeLock A30 can securely store the public key required to verify the firmware signature. For verifying that the signature, ECDSA and SHA algorithms are supported.

Meet 2.a, 2.b, 2.c, 3.a: It is advised to avoid using only CRC checks to ensure the integrity of data as it is not secure. EdgeLock A30 can handle the integrity of code/data one of two ways. The first is with the use of use of a MAC such as: HMAC or CMAC. The second is hashing with the following algorithms: SHA 256 and SHA-384.

The following NXP material can be used as a starting point to meet the requirements listed above:

Table 19. NXP material: Integrity functions

NXP material	Relevant content
NX middleware APIs	• **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish() • **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_finish() • **ECDH Key Derivation:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2(), sss_derive_key_one_go() • **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()
NX middleware demos and examples	• **Symmetric AES Encryption example (key injection):** ex_symmetric • **HMAC Example:** ex_hmac • **Message Digest example:** ex_md • **ECDH Key Derivation example** ex_ecdh • **ECC Signing/Verifing example (key generation):** ex_ecc
Application notes	• AN14238 Get started with EdgeLock A30 support package • AN14559 Migration Guide from EdgeLock A5000 to EdgeLock A30

5.1.5 Confidentiality

Support should be in place to allow the confidentiality of all data which could lead to patient harm. If the authentication and authorization of the system has been implemented correctly, confidentiality is mostly assured. The EdgeLock A30 can encrypt data before it is transmitted into the network to safeguard it from being intercepted. The EdgeLock A30 is Common Criteria EAL 6+ certified (OS including IoT application). Refer to the guidance laid out in Section 4.1.1 and Section 4.1.2 for guidance on implementing confidentiality.

Support should be in place to allow the confidentiality of all data which could lead to patient harm. If the authentication and authorization of the system has been implanted correctly, confidentiality is mostly assured. The EdgeLock A30 can encrypt data before it is transmitted into the network to safeguard it from being intercepted. Refer to the guidance laid out in Section Section 4.1.1, Section 4.1.2, and Section 4.1.3 for guidance on implementing confidentiality.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.1.6 Event detection and logging

In the case of an attack, it is vital for the system to have measures in place for detecting the attack and logging the necessary data. To assist with detecting an attack, all the previously mentioned authentication and integrity checks can be used in identify if there is an attempted attack. Once an attack has been detected the system must keep a log for it to be investigated at a later time. The FDA requirement is as follows:

1. The device should be able to securely create and store log files off the device to track attempted attacks.

Event detection and logging are more linked to being a system function. However, there are ways in which the EdgeLock A30 can be used in conjunction with the host to ensure that the data can be logged securely.

Meet 1: Once an attack is detected EdgeLock A30 can log and be used to securely log data into the secure memory (up to 16 kB of user memory). To ensure further security of the data, EdgeLock A30 can ensure the confidentiality and integrity of the data by encrypting, MACing/hashing the data. This can then allow the device to send data via an unsecure network to keep data off the device.

EdgeLock A30 supports to establish an authenticated session to ensure the connection between host and the A30 is authentic, confidential, and has integrity.

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 an **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.

Table 20. NXP material: Event detection and logging functions

NXP material	Relevant content
Plug & Trust middleware APIs	• **Establish a secure secure messaging channel:** sss_nx_session_open() • **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish() • **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_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()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption example (key injection):** ex_symmetric • **ECC Signing/Verifing example (key generation):** ex_ecc • **HMAC Example:** ex_hmac • **Message Digest example:** ex_md

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.1.7 Resiliency and recovery

Devices can often come under attack. It is critical, especially in healthcare applications, that these devices continue to work as intended. Devices need to be able to maintain function while under attack. The recommendation to achieve this “cyber-resilience” is as follows:

1. Design devices to provide methods for retention and recovery of trusted default devices configurations by an authenticated, authorized user.

Meet 1: Once the EdgeLock A30 has been setup, and the keys injected all crypto algorithms can still be performed once a connection has been established and communication has not been interrupted. The physical connection to the SA will be via the I2C interface. This can ensure that even if taken offline the device can still authenticate users. If the communication with the cloud has been severed EdgeLock A30 would be able to securely store data in the secure storage.

Listed below is material from NXP that can help meet the requirement set above:

Table 21. NXP material: Resiliency and recovery functions

NXP material	Relevant content
NX middleware APIs	• **Establish a secure secure messaging channel:** sss_nx_session_open() • **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish() • **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_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()
NX middleware demos and examples	• **Symmetric AES Encryption example (key injection):** ex_symmetric • **ECC Signing/Verifing example (key generation):** ex_ecc • **HMAC Example:** ex_hmac • **Message Digest example:** ex_md

5.1.8 Firmware and software updates

Devices should be able to be updated securely and quickly. Beyond this, the FDA also recommends manufacturers also plan for rapid testing, evaluation, and patching of devices once in the field. The recommendations are as follows:

1. Design devices to allow firmware and software patches.
2. Implement a secure process for providing validated software updates and patches for users.

As previously mentioned in Section 4.1.4, there are two sections to the system. The EdgeLock A30 and the host. The EdgeLock A30 being CC 6+ compliant can ensure with its integrity protection that only valid updates will be accepted. However, the host must use the EdgeLock A30 in order to ensure the updates are compliant with the standard.

Meet 1, 2: EdgeLock A30 can securely store the ECC private key that is required to decrypt a firmware update that has been encrypted using the associated public key. The private key is protected in the SA tamper-resistant hardware and never leaves the boundaries of the SA secure environment. Symmetric encryption / decryption is supported as well using AES-128/256 (ECB, CBC, CTR, GCM and CCM). EdgeLock A30 can securely store the public key required to verify the firmware signature. The public key can be protected from deletion and overwriting (or other unintended usage) using secure object policies. For verifying the signature, ECDSA and SHA algorithms are supported.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 22. NXP material: Secure firmware update

NXP material	Relevant content
Plug & Trust middleware APIs	• **Establish a secure secure messaging channel:** sss_nx_session_open() • **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish() • **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_finish() • **ECDH Key Derivation:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2(), sss_derive_key_one_go() • **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()
Plug & Trust middleware demos and examples	• **Symmetric AES Encryption example (key injection):** ex_symmetric • **ECC Signing/Verifing example (key generation):** ex_ecc • **ECDH Key Derivation example** ex_ecdh • **HMAC Example:** ex_hmac • **Message Digest example:** ex_md

5.2 Meeting MDR requirements with EdgeLock A30

Within Section 3.3 of the MDCG 2019/16, an indicative list of security capabilities has been shared to ensure the device security capabilities. The share list should be used to help guide the design process of medical devices. A more concise list:

- Authentication
- Authorization
- Confidentiality
- Integrity
- Configuration of Security Features

This section focuses on how EdgeLock A30 can be used to facilitate compliance with some of the security capabilities outlined in *MDCG 2019-16*

5.2.1 Authentication

Authentication is referred to in three separate requirements listed in the MDCG 2019/16:

1. The authenticity of a node.
2. Personal data authenticity.
3. Person authentication.

For 1, 2, 3: EdgeLock A30 supports the cryptographic requirements and operations defined by TLS v1.2/ v1.3 using ECC keys. For ECC keys, both the ECDHE algorithm for key agreement and ECDSA algorithm for digital signatures are supported. AES symmetric keys of up to 256 bits are supported and can be used in ECB, CBC, CTR, GCM and CCM operation modes for data encryption. Hashing algorithms such as SHA-256 and SHA-384 are supported as well. To simplify the integration of TLS, the NX middleware an implementation which allows mbedTLS stack to use the secure element to perform the crypto operations that are part of the TLS handshake between client and server. Such an plugin is as well available for OpenSSL and PKCS#11. Within the EdgeLock A30 keys for both asymmetric and symmetric can be stored. There is a secure user memory available that can be used to store credentials. The following NXP material can be used as a starting point to meet the requirements listed above:

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 23. NXP material authentication

NXP material	Relevant content
NX middleware APIs	• **Read certificate / binary data / certificate:** sss_key_store_get_key (), nx_ReadData() • **Inject a certificate / binary data / certificate:** sss_key_store_set_key (), nx_WriteData() • **Key creation:** sss_key_store_generate_key () • **Key import / injection:** sss_key_store_set_key(), nx_ManageKeyPair(), nx_ChangeKey() • **ECDH Key Derivation:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2(), sss_derive_key_one_go() • **Read EdgeLock A30 pre-injected UID:** nx_GetCardUID()
NX middleware demos and examples	• **OpenSSL Engine/Provider: TLS Client example** • **AWS Demo for Raspberry Pi** • **AWS Cloud Demo on FreeRTOS** • **MCXN-947 AWS Cloud Demo** • **Symmetric AES Encryption example (key injection):** ex_symmetric • **ECC Signing/Verifing example (key generation):** ex_ecc • **ECDH Key Derivation example** ex_ecdh • **Get UID example:** ex_get_uid
Application notes	• AN14238 Get started with EdgeLock A30 support package • AN14559 Migration Guide from EdgeLock A5000 to EdgeLock A30

5.2.2 Authorization

Authorization is defined as the access rights entities within the system have to other parts of the system. This could be a device in the system needing to be authorized to receive data from the server. A well defined and implemented authorization scheme would assign entities the minimum required level for the entity to perform all tasks. Examples of entities that need authorization are as follows:

1. Authorization of users.
2. Authorization of devices.

Meet 1, 2: EdgeLock A30 can employ the use of UIDs to enable authorizations, each UID would have a certain level of access. These UID would be attributed to each user or entity such as another device. Within the backend of the system the access rights can be stored. When establishing a connect, the entity would need to communicate its UID and provide some sort of integrity such as signing. This can then be verified by the EdgeLock A30. The following NXP material can be used as a starting point to meet the requirements listed above:

Table 24. NXP material: authorization

NXP material	Relevant content
NX middleware APIs	• **Read EdgeLock A30 pre-injected UID:** nx_GetCardUID() • **Sign / 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()
NX middleware demos and examples	• **Get UID example:** ex_get_uid • **ECC Signing/Verifing example:** ex_ecc

5.2.3 Confidentiality

It is vital to ensure that information within the system, whether it is stored or transmitted, is confidential. Outlined within the MDCG 2019/16 are three capabilities specifically for confidentiality. They are as follows:

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

1. Personal data de-identification
2. Personal data storage confidentiality
3. Transmission confidentiality

Meet 1,2,3: EdgeLock A30 supports the cryptographic requirements using ECC keys. EdgeLock A30 securely stores the private keys. The ECC curves supported include NIST P-256 and Brainpool P256r1. MACing (HMAC, CMAC), hashing (SHA-256 and SHA-384), TRNG compliant to NIST SP800-90B, and AIS31 class PTG.2. Symmetric encryption and decryption is also supported using AES. AES symmetric keys of 128 and 256 bits are supported and can be used in ECB, CBC, CTR, GCM and CCM operation modes for data encryption. The following NXP material can assist in meeting the confidentiality requirements:

Table 25. NXP material: authorization

NXP material	Relevant content
NX middleware APIs	• **Key creation:** sss_key_store_generate_key () • **Key import / injection:** sss_key_store_set_key(), nx_ManageKeyPair(), nx_ChangeKey() • **ECDH Key Derivation:** sss_derive_key_dh_one_go(), sss_derive_key_dh_two_step_part1(), sss_derive_key_dh_two_step_part2(), sss_derive_key_one_go() • **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish()
NX middleware demos and examples	• **Symmetric AES Encryption example (key injection):** ex_symmetric

5.2.4 Integrity

The methods to ensure the integrity of data change depending on the cryptographic method chosen. The importance of ensuring integrity is to ensure the data arriving to the device has not been modified and is the correct data. The two capabilities highlighted in the MDCG/2019/16 are:

1. Personal data integrity
2. Transmission integrity

It is important to note that the integrity is separated into 2 separate sections. There is the integrity of the EdgeLock A30. This has been ensured with the CC 6+ certification. The second part is the integrity of the host. In this second part the EdgeLock A30 would be used with the host to ensure the integrity.

Meet 1, 2: EdgeLock A30 can handle the integrity of code/data one of two ways. The chosen cryptography method determines the required operation for integrity. The first is the with the use of a MAC such as: HMAC and CMAC. The second is hashing with the following algorithms: SHA 256 and SHA-384.

Table 26. NXP material: integrity functions

NXP material	Relevant content
NX middleware APIs	• **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_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()
NX middleware demos and examples	• **HMAC Example:** ex_hmac • **Message Digest example:** ex_md • **ECC Signing/Verifing example (key generation):** ex_ecc
Application notes	• AN14238 Get started with EdgeLock A30 support package

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 26. NXP material: integrity functions*...continued*

NXP material	Relevant content
	• AN14559 Migration Guide from EdgeLock A5000 to EdgeLock A30

5.2.5 Configuration of security features

It is vital while a system is being developed that the security configurations such as keys used or authorization permissions can be molded to fit the use case better. EdgeLock A30 can enable this flexibility as the developer is able to customize keys and credentials beyond the scope of the Ease-of-Use configuration. This can be done through the process of adding secure objects, creating crypto object, and more. Listed below are NXP material recommendations to assist:

Table 27. NXP material: Configuration of security features

NXP material	Relevant content
NX middleware APIs	• **Symmetric encryption and decryption operations:** sss_nx_cipher_one_go(), sss_nx_cipher_init(), sss_nx_cipher_update() and sss_nx_cipher_finish() • **MACing operation:** sss_mac_one_go(), sss_mac_init(), sss_mac_update(), sss_mac_finish() • **Hashing operations:** sss_digest_one_go(), sss_digest_init(), sss_digest_update(), sss_digest_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()
NX middleware demos and examples	• **NX Personalization Example** nx_Personalization • **NX-CLI Tool** nxclitool
Application notes	• AN14238 Get started with EdgeLock A30 support package • AN14559 Migration Guide from EdgeLock A5000 to EdgeLock A30

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6 References

[1] Document – FDA-2021-D-1158 – Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions (link)
[2] Document – 32017R0745 – Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices (link)
[3] Document – MDCG 2019-16 – Guidance on Cybersecurity for medical devices (link)
[4] Data sheet – SE050 – Plug & Trust Secure Element (link)
[5] Data sheet – SE051 – Plug & Trust Secure Element (link)
[6] Data sheet – SE052 – Plug & Trust Secure Element (link)
[7] Data sheet – A5000 – EdgeLock Secure Authenticator (link)
[8] Application note – AN13030 – Plug & Trust MW Documentation (link)
[9] Data sheet – A30 – Secure Authenticator (link)
[10] Middleware – EdgeLock NX middleware (link)

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7 Revision history

Table 28. Revision history

Document ID	Release date	Description
AN14252 v.2.0	5 December 2025	Editorial changes (typos, etc.). • Section 1.2 "Introducing NXP secure solutions for MIoT": updated. • Section 2 "Architecture of an IoMT System": updated section title, replaced "MIoT" with "IoMT" throughout this section. • Section 2.1 "Security considerations for IoMT": updated section title, replaced "MIoT" with "IoMT" throughout this section. • Section 3 "Regulations for IoMT devices": updated section title, replaced "MIoT" with "IoMT" throughout this section. • Section 4 "Ease MDR and FDA cyber security requirements with EdgeLock SE05x secure elements and A5000": updated section title. • Section 4.1 "Meeting FDA requirements with EdgeLock SE05x/A5000": updated section title (previously "FDA requirements"). • Section 4.2 "Meeting MDR requirements with EdgeLock SE05x/A5000": updated section title (previously "MDR requirements"). • Section 5 "Ease MDR and FDA cyber security requirements with EdgeLock A30": added. • Section 6 "References": updated.
AN14252 v.1.0	28 August 2024	• Initial version

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

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.

EdgeLock — is a trademark of NXP B.V.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Tables

Tab. 1. FDA: security control categories 9
Tab. 2. MDR: security practices 10
Tab. 3. NXP material: Authentication 14
Tab. 4. NXP material: Authorization functions 14
Tab. 5. NXP material: Cryptographic functions 15
Tab. 6. NXP material: Integrity functions 16
Tab. 7. NXP material: Event detection and logging functions 17
Tab. 8. NXP material: Event detection and logging functions 17
Tab. 9. NXP material: Secure firmware update 18
Tab. 10. NXP material: Authentication 19
Tab. 11. NXP material: Authorization 20
Tab. 12. NXP material: Confidentiality 21
Tab. 13. NXP material: Integrity functions 21
Tab. 14. NXP material: Cybersecurity Product upgrades 22
Tab. 15. NXP material: Configuration of security features 22
Tab. 16. NXP material: Authentication 25
Tab. 17. NXP material: Authorization functions 25
Tab. 18. NXP material: Cryptographic functions 26
Tab. 19. NXP material: Integrity functions 27
Tab. 20. NXP material: Event detection and logging functions 28
Tab. 21. NXP material: Resiliency and recovery functions 29
Tab. 22. NXP material: Secure firmware update 30
Tab. 23. NXP material authentication 31
Tab. 24. NXP material: authorization 31
Tab. 25. NXP material: authorization 32
Tab. 26. NXP material: integrity functions 32
Tab. 27. NXP material: Configuration of security features 33
Tab. 28. Revision history 35

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Figures

Fig. 1. Example of how to integrate NXP EdgeLock secure elements and secure authenticators into an MIoT device 4
Fig. 2. Architecture of a typical IoMT system 5
Fig. 3. MDR regulation structure 10
Fig. 4. SE05x features mapping to FDA and MDR cybersecurity requirements 12
Fig. 5. FDA: cybersecurity requirements overview 13
Fig. 6. EdgeLock A30 features mapping to FDA and MDR cybersecurity requirements 23
Fig. 7. FDA: cybersecurity requirements overview 24

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Contents

1 Introduction to medical IoT 2
1.1 Overview of FDA and EU MDR cybersecurity regulations 2
1.2 Introducing NXP secure solutions for MIoT 3
2 Architecture of an IoMT System 5
2.1 Security considerations for IoMT 6
3 Regulations for IoMT devices 8
3.1 FDA regulation 8
3.2 MDR regulation 9
4 Ease MDR and FDA cyber security requirements with EdgeLock SE05x secure elements and A5000 12
4.1 Meeting FDA requirements with EdgeLock SE05x/A5000 12
4.1.1 Authentication 13
4.1.2 Authorization 14
4.1.3 Cryptography 14
4.1.4 Code, data, and execution integrity 15
4.1.5 Confidentiality 16
4.1.6 Event detection and logging 17
4.1.7 Resiliency and recovery 17
4.1.8 Firmware and software updates 18
4.2 Meeting MDR requirements with EdgeLock SE05x/A5000 19
4.2.1 Authentication 19
4.2.2 Authorization 20
4.2.3 Confidentiality 20
4.2.4 Integrity 21
4.2.5 Cybersecurity product upgrades 22
4.2.6 Configuration of security features 22
5 Ease MDR and FDA cyber security requirements with EdgeLock A30 23
5.1 Meeting FDA requirements with EdgeLock A30 23
5.1.1 Authentication 24
5.1.2 Authorization 25
5.1.3 Cryptography 25
5.1.4 Code, data, and execution integrity 26
5.1.5 Confidentiality 27
5.1.6 Event detection and logging 28
5.1.7 Resiliency and recovery 29
5.1.8 Firmware and software updates 29
5.2 Meeting MDR requirements with EdgeLock A30 30
5.2.1 Authentication 30
5.2.2 Authorization 31
5.2.3 Confidentiality 31
5.2.4 Integrity 32
5.2.5 Configuration of security features 33
6 References 34
7 Revision history 35
Legal information 36

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

© 2025 NXP B.V. **All rights reserved.**