AN13254

Secure attestation with EdgeLock SE05x

Rev. 1.1 — 26 July 2023 Application note

Document information

Information	Content
Keywords	EdgeLock SE05x, attestation, secure element
Abstract	This application note describes what secure attestation is and why it is important to ensure trust in IoT devices. It explains how to use EdgeLock SE05x to attest keys, credentials, and data that resides in the secure element and how this can be used to protect IoT devices against man-in-the-middle attacks, data forge and counterfeiting, among other attacks.

Revision history

Revision history

Revision number	Date	Description
1.1	20230726	• Updated legal information in Section 5 • Updated attestation example flow diagram in Figure 7
1.0	20230621	First release

1 Introduction

The use of small embedded systems and IoT devices for collecting, processing, and transferring security-critical data is growing significantly in many different applications: from industrial control systems and vehicular systems, to home equipment and automation systems, just to mention a few examples.

At the same time, the recent increase in cybersecurity threats such as replay attacks, man-in-the-middle attacks, malicious code modification and counterfeiting undermines the level of trust in connected devices and in data transiting through the network.

Consider an OEM who operates a network of sensors. This network is composed of different types of sensors which are distributed in the field and are not subject to any direct human interaction or supervision. Sensors generate measurement data, which is periodically transferred to the OEM backend or cloud service. This service analyzes the data collected from the different sensors and triggers certain actions based on the collected measurements. In such a scenario, transmitted sensor data might become a target for man-in-the-middle attacks as shown in Figure 1. In fact, attackers may try to tamper with the data while it is transiting through the network or they can try to impersonate the device identity and send fake data to the remote service. Counterfeited devices could also be deployed without the OEM's knowledge and exploit or disrupt the OEM's network and services.

Figure 1. IoT device data is sensitive to man-in-the-middle attacks

To mitigate these threats, secure attestation procedures supported by hardware-backed cryptographic keys and algorithms are needed. With secure attestation in place, it is possible to prove to an external service that a piece of information, be it the device identity or some other data generated by the device, originated from or passed through a trusted source and that such information was not tampered with by malicious actors.

A highly effective way of implementing secure attestation is by integrating a dedicated Secure Element (SE) such as EdgeLock SE05x into the IoT device. In fact, EdgeLock SE05x can be used to establish a root of trust based on chip-unique attestation keys that are kept secure in the IC. The scope of attestation keys stored in EdgeLock SE05x is restricted so that they can only be used for attesting data that originates in the SE.

Moreover, thanks to EdgeLock SE05x I^2C controller interface, it is possible to read and attest data generated by a connected trusted subsystem (for example a sensor). For example, in the IoT device architecture depicted in Figure 2, the sensor data is attested by EdgeLock SE05x before it is transmitted to a remote service. In case a malicious attacker attempts to tamper with the sensor data, the remote service will detect that data is not authentic and would be able to reject the data or take the appropriate countermeasures.

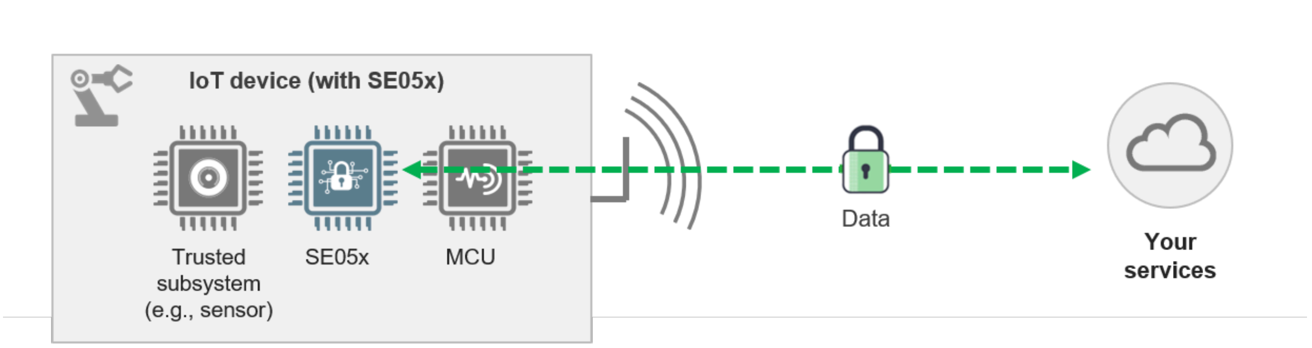

Figure 2. IoT device attestation with EdgeLock SE05x

2 Secure attestation overview

Secure attestation is a means to undeniably prove to a third party that a piece of data has originated in a trusted environment, for example an SE such as EdgeLock SE05x. When data is requested from the SE, the user can request attestation for the returned data. The SE attests the origin of the data by signing it with a chip-unique key-pair (attestation key-pair) that is securely stored inside the SE.

To implement a truly secure attestation process, the attestation key-pair must have the following properties:

- The attestation key-pair must be pre-provisioned by the SE manufacturer in a secure and controlled environment or, alternatively, generated by the OEM inside the SE. This ensures that the private part of the attestation key-pair (`K_ATT_PRIV`) is never exposed outside of the SE and used for any purpose other than attestation.
- The attestation key-pair cannot be used to sign arbitrary data - for example data generated externally by the host device. This is different to a traditional key-pair used for signing. The attestation key-pair is only allowed to sign data that originates in the SE. As such, the scope of an attestation key-pair must be limited so that the SE can ONLY use it to sign data that is stored in the SE.

If the abovementioned conditions are respected, then an external service can verify the signature of the retrieved data using the public part of the attestation key-pair (`K_ATT_PUB`) and be sure that data could only have originated in that particular SE.

For the attestation key-pair to be trusted by a service, a trust-chain must be established either by manually registering `K_ATT_PUB` as a trusted key in the service or by using a certificate chain. In the latter case, the attestation key-pair can be backed by an attestation certificate that holds `K_ATT_PUB` and that is signed by a trusted CA (`CA_CERT`). The attestation certificate can be stored in the SE together with the attestation key-pair. It is then presented together with attested data and used by the service to verify the data.

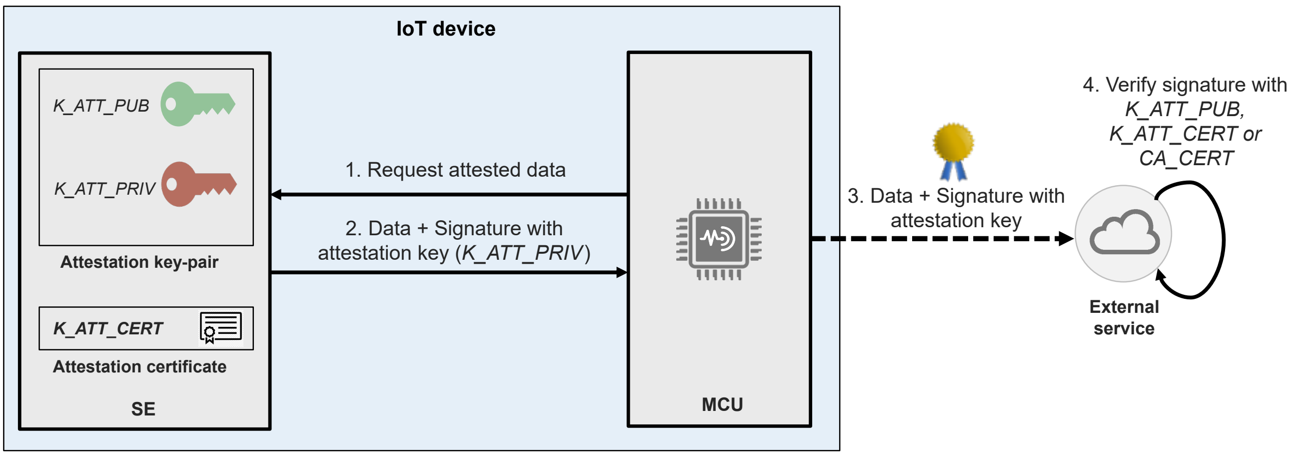

Figure 3. Secure attestation overview

3 Secure attestation with EdgeLock SE05x

EdgeLock SE05x is a ready-to-use secure element solution specifically designed for IoT applications. It provides a root of trust at the IC level and it allows users to generate, provision and manage security credentials and perform cryptographic operations for security critical communication and control functions.

EdgeLock SE05x is security certified to a level of CC EAL 6+ and provides security against physical and logical attacks aimed, for example, at extracting security keys. EdgeLock SE05x supports both RSA and ECC asymmetric cryptographic algorithms with high key length and future proof ECC curves.

Additionally, EdgeLock SE05x is pre-provisioned with keys and credentials in a highly secure and controlled environment. All EdgeLock SE05x variants include an attestation key trust provisioned by NXP. Such key is already configured with the correct policies and restrictions as described in Section 2 so that it can be used out-of-the box to implement secure attestation use cases. The EdgeLock SE05x variants C, E and F also contains an attestation certificate.

Moreover, EdgeLock SE05x comes with a pre-installed IoT applet offering advanced key management and cryptographic functions. It allows users to easily request and obtain attested data, and to generate and configure additional attestation keys if needed.

To ease the integration of the applet functionalities in the IoT solution, EdgeLock SE05x even provides a fully-featured middleware package. The middleware is pre-integrated with many micro-controller platforms and contains several examples and demo projects that can be used as a starting point for custom software implementations, including secure attestation use cases.

This section explains how to leverage EdgeLock SE05x and the EdgeLock SE05x Plug & Trust middleware to implement secure attestation in your solution.

This document focuses on the attestation format and commands used in products using the IoT Applet starting at version 7.2.0, so products like SE050E, SE051A and SE051C.

3.1 Read secure objects with attestation

Keys and credentials are stored in EdgeLock SE05x as secure objects. Each secure object contains, besides the value of the secure object - like for example a private-public key-pair or a binary file - a set of attributes which beside other metadata include the secure object ID, the policies with access rules for the secure object and the origin of the secure object. The origin tells if the object got generated internally, externally, or was trust-provisioned by NXP. Secure objects and their attributes can be read by the IoT device at any time. Only the public part of the secure object is returned (for example the public key of an asymmetric key-pair), while the private part remains secure in the SE.

EdgeLock SE05x allows devices to read secure objects with attestation. In this case, secure objects and their attributes are returned together with a signature over the full payload of the response using the specified attestation key and algorithm.

Figure 4 summarizes the typical attestation flow using EdgeLock SE05x:

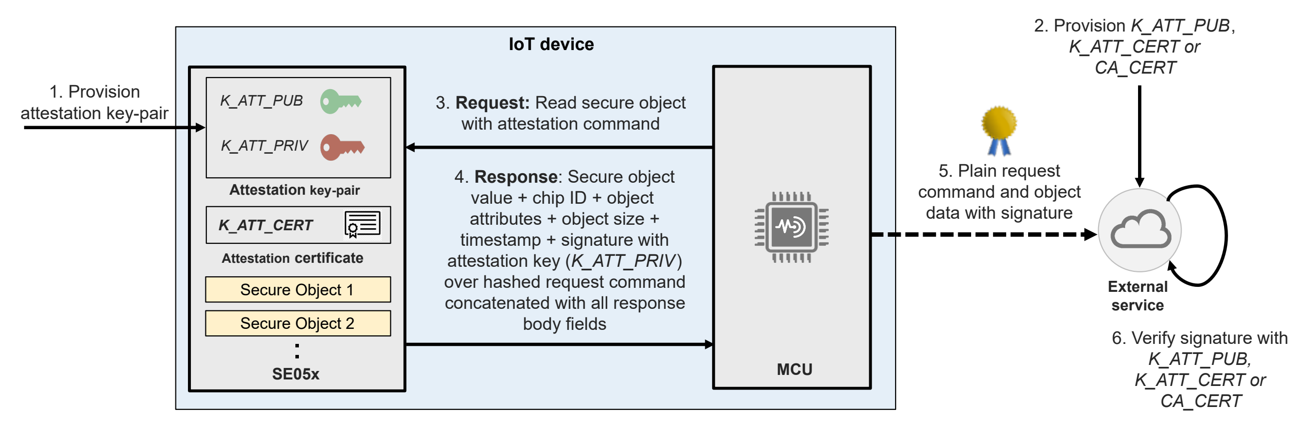

Figure 4. Secure attestation flow with EdgeLock SE05x

1. An attestation key-pair (`K_ATT`) must be provisioned in EdgeLock SE05x. All EdgeLock SE05x variants come with a pre-provisioned attestation key that can be used out-of-the-box (see AN12436 or AN12973 for a list of pre-provisioned keys respectively for EdgeLock SE050 and EdgeLock SE051). Alternatively, OEMs can setup their own attestation key-pair by generating an RSA or ECC key-pair (depending if the type supports RSA or ECC or both) in EdgeLock SE05x and by setting the correct object policy. For secure attestation, the key-pair must have at least the policy access rules shown in Table 1. Optionally, OEMs can provision an attestation certificate (`K_ATT_CERT`) signed by a trusted CA and holding the public part of the attestation key-pair (`K_ATT_PUB`).
   Note: *EdgeLock SE05x variants C,E and F includes a pre-provisioned certificate associated to the attestation key that is signed by the NXP Root of Trust entity.*

   Table 1. Required policy rules for attestation keys

Policy rule name	Value	Description
POLICY_OBJ_ALLOW_SIGN	0	Prevent attestation key-pair from being used for normal signing operations.
POLICY_OBJ_ALLOW_DECRYPT	0	Prevent attestation key-pair from being used for decryption operations (for RSA).
POLICY_OBJ_ALLOW_ATTESTATION	1	Allow usage of key-pair for attestation operations.

   Note: *POLICY_OBJ_ALLOW_SIGN = 0 and POLICY_OBJ_ALLOW_DECRYPT = 0 attributes are mandatory and must be added to implement a fully secure attestation.*
2. `K_ATT_PUB` or `K_ATT_CERT` should be provisioned in the service (or services) that will need to verify the attested data - for example a cloud service. In real-world applications where scalability is a concern, the CA certificate (`CA_CERT`) used to sign `K_ATT_CERT` is uploaded to the external service instead of multiple device certificates. This allow the service to verify attested data for a group of devices using a single trusted certificate (`CA_CERT`).
3. The IoT device MCU reads a secure object with attestation from EdgeLock SE05x. The request must contain the reference to the attestation key object, the signature algorithm to apply (see Table 2) and some random freshness data.
   Note*: only secure objects with attribute POLICY_OBJ_ALLOW_READ = 1 can be read with attestation. The attributes of a secure object are always signed and returned, even if the secure object has no public part that can be read (for example a symmetric key). This can be useful to verify the state of the object (check the origin attribute).*

Table 2. Supported signing algorithms

Key type	Algorithm	Digest input
RSA	RSASSA-PSS	SHA-224, SHA-256, SHA-384, SHA-512
	RSASSA-PKCS1	SHA-224, SHA-256, SHA-384, SHA-512
EC (NIST, Brainpool curves)	ECDSA	SHA-224, SHA-256, SHA-384, SHA-512
EC (Edward curves)	EDDSA	SHA-512

4. The SE retrieves the secure object and prepares the response. The response is sent to the MCU and contains:
   - Data read from the secure object (if not secret)
   - Chip unique identifier
   - Secure object attributes
   - Secure object size
   - Timestamp (a monotonic counter value)
   - Signature over the command and response

   The signature is calculted over the hash of the full plain (unencrypted) request command (except the ISO 7816 *Le* bytes at the end) concatenated with the secure object value (public part, if any), the chip unique identifier, the object attributes, the object size and the timestamp as returned in the response (including type and length for each field). The signature is performed using the attestation key and coresponding algorithm specified in the request.
   Note: *Hashing of the plain request command is performed using a bit size matching the strength of the attestation algorithm (for example SHA384 for SIG_ECDSA_SHA_384).*
5. The response received from the SE (containing the corresponding signature) is sent to an external service. The request command to the SE must be sent together with the response so that the external service can compute the hash required to verify the signature provided in the response.
6. Before accepting the data, the external service verifies the signature using `K_ATT_PUB`, `K_ATT_CERT` or `CA_CERT`. If the signature is valid, then the service can be sure that data was indeed retrieved from EdgeLock SE05x and that it was not manipulated after being retrieved from the secure element. To avoid replay attacks, the timestamp and freshness fields must also be checked for each attestation to prevent reuse of attestation. In particular it need to be checked that:
   - The timestamp field for consecutive attestations MUST be consecutive.
   - The freshness field for consecutive attestations MUST be different.

3.2 Attestation of generated key-pairs

Public Key Infrastructure (PKI) is widely used to manage identities and security in IoT deployments. In such a scenario, IoT devices might need to generate key-pairs and upload the corresponding public keys to an external service or another host. Nevertheless, as anyone can potentially generate key-pairs, the public key integrity and authenticity must be guaranteed in order for the service to be able to trust any cryptographic operation done with it (for example verifying a digital signature).

EdgeLock SE05x allows devices to generate key pairs on-chip, so that the private key counterpart remains secure and never leaves the IC secure storage. Leveraging on this feature, a remote service might request a device to create a key-pair in EdgeLock SE05x and to transfer the corresponding public key to the service so it can be registered. To prove to the remote service that the public key was not manipulated, that it comes from a trusted device and not from a third party or attacker and that the corresponding private key actually resides in the SE, the device host MCU can request EdgeLock SE05x to sign the public key with the attestation key before sending it to the remote service. The service will only accept the public key if the signature can be verified with the attestation public key.

Figure 5 shows a simplified representation of the flow when reading a key-pair with attestation from EdgeLock SE05x.

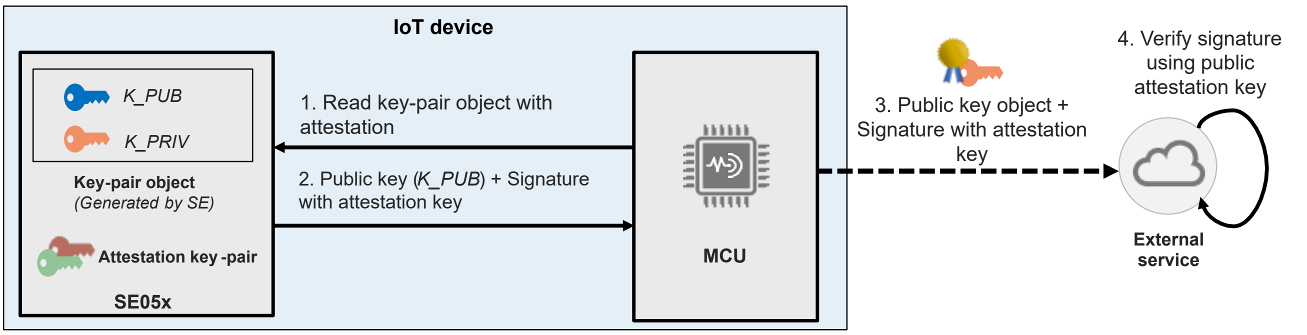

Figure 5. Attestation of generated key-pairs with EdgeLock SE05x

3.3 Attestation of data read through I²C

Handling unprotected data collected by attached sensors or actuators directly in the host MCU might expose data to potential security threats. In fact, data can be accessed and manipulated more easily when there are no cryptographic measures to protect it. To overcome this issue, EdgeLock SE05x allows users to connect a secondary IC (for example a sensor) through its additional I²C interface. In this configuration, EdgeLock SE05x acts as I²C controller, while the sensor node operates as follower in the I²C bus. Communication with the connected IC to write / read data is only allowed through EdgeLock SE05x interface (optionally after authentication is performed), therefore adding an additional layer of protection for sensitive data. For more information on EdgeLock SE05x I²C capabilities, please refer to AN12449.

For additional security, EdgeLock SE05x allows users to attest the data retrieved from the I²C interface using attestation keys before data is passed to the host MCU. In this way, it is possible to prove to an external service (for example a remote cloud service) that data was indeed collected from a trusted EdgeLock SE05x SE and that no manipulation occurred after data was retrieved from the secure IC. A simplified representation of the process is shown in Figure 6.

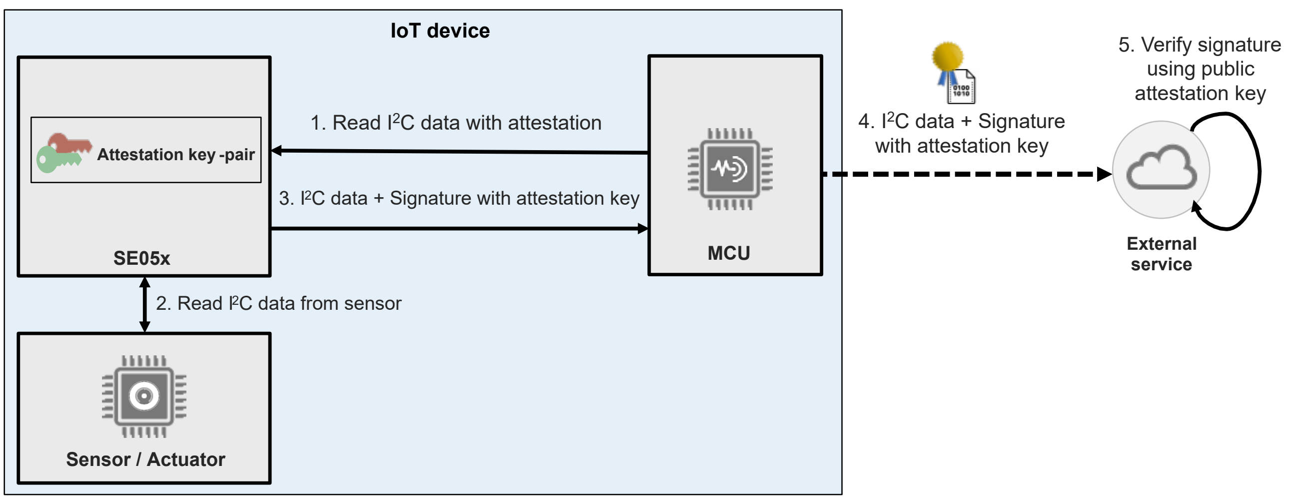

Figure 6. Attestation of sensor data with EdgeLock SE05x

4 EdgeLock SE05x Plug & Trust middleware attestation examples

To ease the integration of the IoT applet functionalities in the IoT solution, EdgeLock SE05x provides a fully-featured Plug & Trust middleware package. The middleware is pre-integrated with many micro-controller platforms and contains several examples and demo projects that can be used as a starting point for custom software implementations.

IoT applet version starting at 7.2.0 implements a new attestation command format. Depending on the selected IoT applet version at compilation time (C-define `PTMW_SE05x_Ver`) the MW uses the corresponding format.

In the context of secure attestation, the available examples are listed in Table 3.

Table 3. Attestation examples in EdgeLock SE05x Plug & Trust middleware

Name	Source code path	MW documentation path
Read object with attestation	`/simw-top/demos/se05x/se05x_ReadWithAttestation/se05x_ReadWithAttestation.c`	`/simw-top/doc/demos/se05x/se05x_ReadWithAttestation/Readme.html`
Read ECC NIST key with attestation	`/simw-top/sss/ex/attest_ecc/ex_sss_ecc_attest.c`	`/simw-top/doc/sss/ex/attest_ecc/readme.html`
Read Montgomery key with attestation	`/simw-top/sss/ex/attest_mont/ex_sss_mont_attest.c`	`/simw-top/doc/sss/ex/attest_mont/readme.html`
Read I²C data with attestation	`/simw-top/demos/se05x/se05x_I2cMaster/se05x_I2cMasterWithAttestation.c`	`/simw-top/doc/demos/se05x/se05x_I2cMaster/readme.html`

- **Read object with attestation:** this example demonstrates how to read a generic secure object with attestation and parse the attested data to check various object attributes. In the example, an ECC NIST P-256 key-pair is used as the attestation key and a binary object is attested. First, an attestation key is created using the *sss_key_store_generate_key ()* function and a policy with attribute *can_Attest = 1* is assigned to the key. Then, a sample binary secure object is created using *sss_key_store_set_key()* function. Finally, the binary object is read with attestation using the *sss_se05x_key_store_get_key_attst()* function. The function takes as input parameters a handle to the ECC NIST P-256 attestation key, the algorithm to use for the signature (ECDSA with SHA-256) and some randomly generated freshness data. For demonstration purposes, the verification of the attested data is performed in the host. The *sss_digest_one_go ()* is used to compute the hash of the data and *sss_asymmetric_verify_digest()* is used to verify the attestation signature. The flow of the example is schematized in Figure 7.
  Note*: the code also includes an example implementation of how to read large binary secure objects with attestation by calling Se05x_API_ReadObject_W_Attst() multiple times.*
- **Read keys with attestation**: the EdgeLock SE05x Plug & Trust middleware provides two examples that showcase how to read a public key with attestation. The first example shows how to read with attestation an ECC NIST P-256 public key, while the other example shows how to read a Montgomery 25519 public key. Both examples use an ECC NIST P-256 attestation key to sign the data using the ECDSA algorithm with SHA-256.
- **Read I²C data with attestation:** this demo example demonstrates how to read data with attestation from a sensor (accelerometer) connected through I²C to EdgeLock SE05x. The demo uses the *Se05x_i2c_master_attst_txn()* function to poll the sensor for new data. The sensor data is read with attestation (ECDSA signature with SHA-512) using a pre-injected attestation key (ECC NIST P-256). Follow the instructions provided in AN12449 for step-by-step instructions on how to run the demo.

The flow of attestation shown in Figure 7 shows the flow of the example "se05x_ReadWithAttestation". The attestation is created using the secure element and being verified on the host.

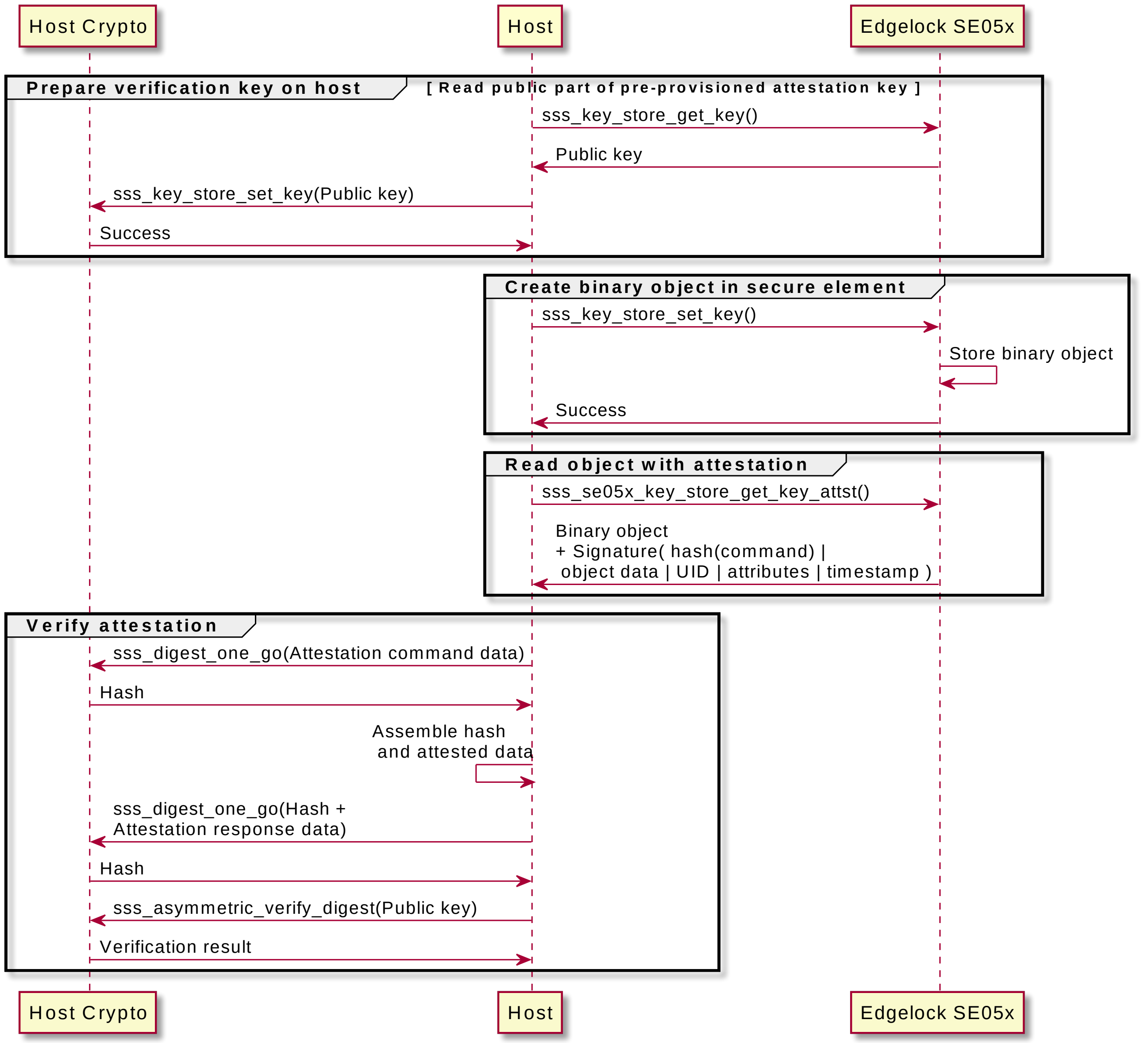

Figure 7. Flow of read object with attestation example

5 Legal information

5.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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

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.

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.

5.3 Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

Tables

Figures

Contents

Date of release: 26 July 2023
Document identifier: AN13254