IoT Secure Element Protection Profile (IoT-SE-PP)

Version 1.0.0, 2019-12-19

developed by
Secure Communications Alliance (SCA),
IoT PP working group:
Shanghai AOH Smart Technology Co., Ltd.
ChengDu JAVEE Microelectronics Co., Ltd.
ESIM Technology Co., Ltd.
FEITIAN Technologies Co., Ltd.
Haier Uplus Intelligent Technology (Beijing) Co., Ltd.
Infineon Technologies AG Co., Ltd.
NXP Semiconductors B.V.
STMicroelectronics
TechKnowledge Services Group Inc.
WuHan TianYu Information Industry Co., Ltd.

Contents

1 PP introduction

The purpose of this Common Criteria (CC) Protection Profile (PP) is to standardize the security requirements of an IoT Secure Element (IoT SE) to be used in an IoT device. This PP targets IoT devices, which are home appliances like washing machines, refrigerators, air conditioners, etc. Furthermore, the intention behind the IoT SE is to support an IoT Secure Communications Module (IoT SCM), which is subject to the separate Protection Profile IoT-SCM-PP. Usage of an IoT SE without an IoT SCM is not recommended, as some aspects of a secure integration of an IoT SE into an IoT device could require an IoT SCM, which is compliant to the IoT-SCM-PP.

The dedicated IoT SE basically shall provide a unique, provable identity for the IoT device it is built in. Furthermore, the IoT SE shall provide secure end-to-end authentication against a remote administrator of the IoT device (e.g., a backend system in an IoT cloud the IoT device is connected to). To do so, the IoT SE securely stores and processes device-individual SE keys, administrator keys and – as a service for the IoT SCM – other keys usable by the IoT SCM, e.g. those needed to establish secure communication channels between the IoT device and remote network devices. Furthermore, the IoT SE contains an entropy source and provides high quality random numbers for use in the IoT device (the IoT SE itself not necessarily creates its own cryptographic keys). The main goal of the IoT SE is the provision of a security anchor being not practical to clone. That security anchor is used for secure end-to-end data exchange with the remote administrator of the IoT device, and it provides a secure environment for storage and processing of keys of the SCM and the IoT application.

This document is intended to provide a detailed description of the requirements for the IoT SE, the implementation of a concrete solution remains a subject of the IoT SE developer. This PP also does not contain concepts how to use the IoT SE in certain applications, i.e. the functional interface of the IoT SE is not specified by this PP.

Besides from the required functionality and assumptions about its integration into the IoT host device, this PP does not restrict form factors or internal architecture of the IoT SE. As the requirements also include protection against physical attacks, information-leakage analysis and fault-injection techniques, it is assumed that only an IoT SE consisting of dedicated hardware and firmware contained therein can fulfil all requirements of this PP.

As stated before, a TOE evaluated and certified according to this PP, i.e. the IoT Secure Element (IoT SE), is intended to support an evaluated and certified IoT Secure Communication Module (IoT SCM) as specified by the separate Protection Profile IoT-SCM-PP. One of the main goals of the IoT-SCM-PP is to define requirements how an IoT SCM provides secure communication with the network devices it connects to. The hardware of the IoT SE may be shared by the IoT SCM (and even the IoT application) for a higher level of integration, e.g. in terms of a system on chip (SoC).

1.1 PP reference

Title:	IoT Secure Element Protection Profile (IoT-SE-PP)
Version:	1.0.0
Date:	2019-12-19
CC version used:	3.1 Revision 5
Assurance:	EAL4 augmented with AVA_VAN.4 and optionally ALC_FLR.1
PP registration:	BSI-CC-PP-0109, registered by the German Bundesamt für Sicherheit in der Informationstechnik (BSI)
PP authors:	Secure Communications Alliance (SCA), IoT PP working group: Shanghai AOH Smart Technology Co., Ltd. ChengDu JAVEE Microelectronics Co., Ltd. ESIM Technology Co., Ltd. FEITIAN Technologies Co., Ltd. Haier Uplus Intelligent Technology (Beijing) Co., Ltd. Infineon Technologies AG Co., Ltd. NXP Semiconductors B.V. STMicroelectronics TechKnowledge Services Group Inc. WuHan TianYu Information Industry Co., Ltd.

1.2 TOE overview

The TOE type addressed in this PP is an IoT Secure Element (shortly IoT SE or just SE), which is intended to be integrated into an IoT host device and basically providing services – mainly secure key storage, cryptographic operations and random number generation – for the IoT Secure Communications Module (IoT SCM), which is also integrated in the IoT host device. The IoT SE can provide its services not only to the IoT SCM but also to the IoT application, but the latter only via interfaces provided by the IoT SCM. The IoT SE protects the corresponding cryptographic keys during storage in memory and processing in cryptographic operations from disclosure. This protection renders cloning of an IoT SE not practical and thus counters the main threat from perspective of an IoT device admin. The IoT device admin or service provider can be a different party than the IoT device vendor.

The following figure shows the context of the IoT SE TOE. The IoT SE is integrated in the IoT host device together with the IoT SCM and the IoT application. The IoT SE is providing services to the IoT SCM, which is mediating, controlling and protecting any communication of the IoT device with network devices in a WAN (typically the internet), which provide services to the IoT device in the “IoT cloud”. The connection may be direct or mediated by an IoT gateway (which by the way could be an IoT device utilizing an IoT SE and an IoT SCM on its own). The IoT device user may interact with the IoT device indirectly by services provided in the IoT cloud (to control or monitor IoT SE and IoT SCM as far as the cloud-based functionality allows), but they also have a LAN-accessible interface to the IoT SCM, enabling them at least to read the SE ID, the firmware versions of IoT SE, IoT SCM and IoT application, and the network connection rules currently stored in the IoT SCM. The IoT device user also may be a role known by the IoT application and therefore connect directly to the IoT device to control or monitor the IoT application, mediated by the IoT SCM within the connection limits it enforces (as configured by the IoT device admin).

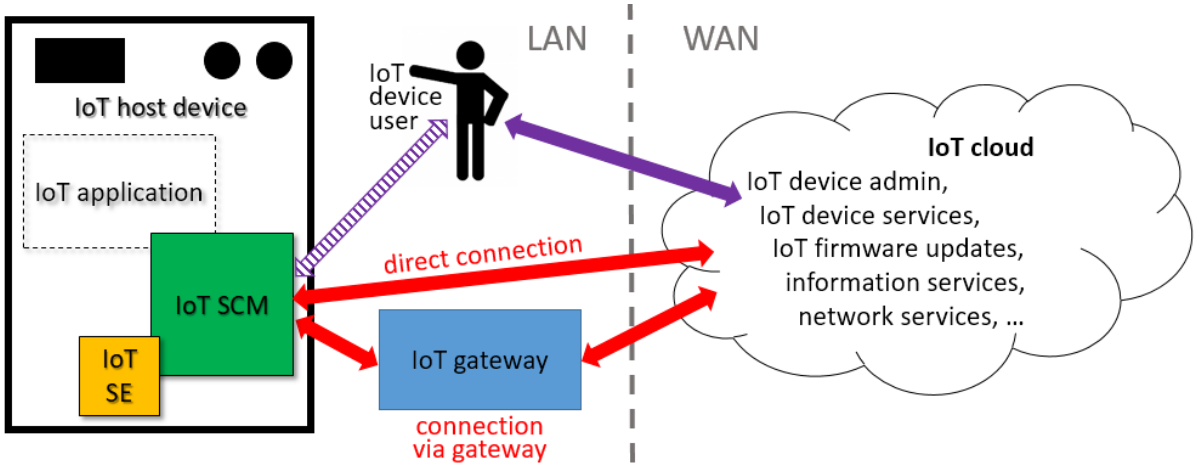


Figure 1: IoT SE TOE in the greater IoT device context

Physical Scope of the TOE

As the IoT SE TOE shall provide a certain level of physical protection for its assets and as it has to include a physical entropy source (either to act as a physical random number generator directly or to produce seed material for a deterministic random number generator), the TOE will have to consist of both, hardware and firmware. The physical form factor of the TOE may be a single integrated circuit, a dedicated secure microcontroller core in a system on chip (SoC), or any other solution that fulfils – among others – the requirements concerning physical protection, information leakage protection and fault-injection resistance.

The TOE, i.e. the IoT SE, is intended to be integrated into an IoT host device including its IoT application, the latter being the IT hardware and firmware of the IoT host device finally making use of the IoT SE to provide a unique identity and authentication features. The IoT host device also integrates an IoT Secure Communications Module (IoT SCM; compare IoT-SCM-PP). Neither IoT application nor IoT SCM belong to the IoT SE TOE by definition, though it might be possible that the physical scopes of IoT SE and IoT SCM or even of IoT SE and IoT SCM and IoT application match or overlap (then both, this PP and the IoT-SCM-PP may be applied on the product integrating IoT SE and IoT SCM, but likely in separated evaluations and certifications due to different assurance requirements in the two PPs).

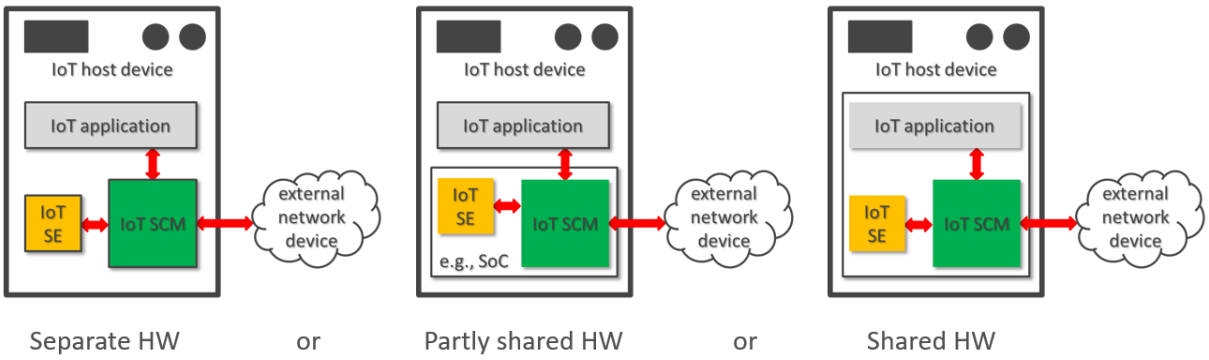


Figure 2: Examples for physical scope of the IoT SE TOE inside the IoT host device

Depending on the concrete form factor of the IoT SE TOE, dedicated evaluation and certification procedures may apply, as defined or adopted by the corresponding certification scheme. For instance, in SOG-IS evaluation and certification schemes, special evaluation requirements may apply according to supporting documents of the Joint Interpretation Library (JIL), e.g. if the TOE would be considered a single security IC or a hardware security box.

Logical Scope of the TOE

The IoT SE TOE shall provide the following security functionality:

- Storage and usage of device-individual cryptographic keys for authentication of the TOE against network devices and remote users and vice versa, and authenticity protection and authenticity verification of data exchanged between the IoT host device via the TOE and connected network devices or remote IoT device users (by adding and verifying signatures/MACs, respectively);
- Access control concerning usage and update of keys stored/used in the TOE;
- Entropy generation and random bit generation as a service for the IoT host device;
- Protection of internally stored assets from disclosure or modification by physical probing, physical modification, information leakage analysis and fault-injection techniques.

Optional Functionality

This PP does not require the following security functionality for the IoT SE TOE, though it might be – among others – added by the ST author:

- Secure update of TOE firmware;
- Generation of cryptographic keys for TOE-internal or TOE-external use (if used, the random number generator of the TOE shall be used as a basis);
- Access-controlled storage as a service functionality for IoT SCM and/or IoT host device (not related to any of the TOE assets currently defined).

For the optional secure update of TOE firmware, the functional package “Secure Update” has been defined in section 7 hereinafter. If the TOE supports firmware update, the functional package “Secure Update” shall be used to model the corresponding part of the SPD, security objectives and security requirements. If a particular IoT SE supported firmware update, but the corresponding ST would not strictly conform (also) to the functional package “Secure Update” as defined in section 7 hereinafter, this shall be deemed as a non-conformance with respect to this PP.

The other optional functionalities listed above are not modelled in this PP in terms of packages. If these or other security functions shall be implemented in the TOE in addition to the definitions in this PP, it is up to the ST author to extend their statements of security problem definition, security objectives and security functional requirements in the ST accordingly. Any additions made must not be in conflict with the definitions in this PP (which would be verified during the evaluation of the ST).

TOE Life-cycle

The life-cycle of the IoT SE TOE can be separated into the following phases:

1. Development of hardware and firmware of IoT SE
2. Production of hardware and firmware of IoT SE
   (with optional integration of IoT SE into IoT SCM)
3. Delivery of completed IoT SE to IoT device manufacturer.
4. Integration of IoT SE (and IoT SCM) into IoT host device
5. Delivery of IoT device to IoT device user
6. Normal operation by IoT device user and IoT admin

Phases 1 to 3 are within responsibility of the IoT SE developer. It shall be ensured that these phases are performed by trusted personnel in secure environments. Since the realization of the phases depend on the concrete SE, it is important that the TOE developer considers and enforces appropriate security measures during phases 1 to 3.
All relevant development, production and delivery sites used in phases 1 to 3 shall be subject to evaluation of assurance aspect ALC.

Phases 4 and 5 are already considered usage phases, which are within responsibility of the IoT device manufacturer. The IoT device manufacturer shall regard the assumptions as stated in section 3.2 hereinafter (as far as these assumptions are applicable, according to the concrete form factor of the IoT SE and the way of integration into the IoT host device).

In phase 3, the certified IoT SE TOE has to be complete and no more modification of the TOE configuration is allowed after that (other than – if supported – updating its firmware with a newer version, which is also certified according to this PP on the same IoT SE hardware).

2 Conformance claims

2.1 CC conformance claim

This PP uses the Common Criteria version 3.1 Revision 5.

This PP is conforming to Common Criteria Part 2 extended.

This PP is conforming to Common Criteria Part 3.

2.2 PP claim and package claim

This PP does not claim conformance to any other PP.

For a TOE not supporting firmware update, this PP is conforming to assurance package EAL4 augmented by AVA_VAN.4 as defined in Common Criteria Part 3.

For a TOE supporting firmware update, this PP is conforming to assurance package EAL4 augmented by AVA_VAN.4 and ALC_FLR.1 as defined in Common Criteria Part 3, and to functional package “Secure Update” as defined in section 7 hereinafter.

2.3 Conformance claim rationale

This PP does not claim conformance to any other PP.

2.4 Conformance statement

This PP requires strict conformance of the ST or PP claiming conformance to this PP. If the TOE as defined by the ST or PP claiming conformance to this PP supports update of its own TOE firmware, strict conformance shall be given only if the ST or PP claiming conformance to this PP is strictly conformant to assurance package EAL4 augmented by AVA_VAN.4 as defined in Common Criteria Part 3 and to functional package “Secure Update” as defined in section 7 hereinafter at the same time.

3 Security problem definition

3.1 Terms and assets

Term	Description
IoT SE (TOE of this PP)	IoT Secure Element, the component in the IoT host device that securely stores and processes persistent cryptographic keys.
IoT SCM	IoT Secure Communication Module, the component in the IoT device that can actually connect to external network devices. Provides services of network connection control and secure channel functionality. The SCM uses the IoT SE to securely store and process persistent cryptographic keys.
IoT host device	A device like e.g. a home appliance that uses the functionality of IoT SE and IoT SCM integrated into that IoT host device.
IoT application	IT part of the IoT host device, which is using services of IoT SE and IoT SCM.
IoT device	Combination of IoT host device inlcuding the IoT application, IoT SE and IoT SCM. An IoT device may be any kind of device that connects to a network (presumably a LAN connected to the internet) and that is able to send or receive information to or from the network or via the network to the internet. IoT devices may communicate with various entities like other IoT devices, IoT gateways and the IoT device admin.
External network device	Any network device external to the IoT device, which the IoT device may connect to via its IoT SCM. May be in the same LAN as the IoT device or in the WAN (i.e. in the IoT cloud).
IoT gateway	A device placed in the same LAN as the IoT device, mediating the connection of the IoT device (and supposedly of other IoT devices in the same LAN) to the IoT device admin or to other network devices in the IoT cloud.
IoT cloud	Sum of all external network devices (clients, servers, etc.) in the WAN, which the IoT device is directly or indirectly connecting to, to send data to or receive data from. IoT device admin is administering the IoT device from the IoT cloud.
IoT device admin	The IoT device admin (administrator) is responsible for the management of the security services of the TOE and the corresponding key management.
IoT device user	The individual who is the actual user of an IoT device, typically its owner or leaseholder. Most of the interaction with the IoT device the IoT user is doing via the IoT cloud, in those cases the IoT device is not aware of the IoT user, but receiving corresponding requests from the IoT admin (on behalf of the IoT device user instead). The Iot device user can read the version information of IoT SE and IoT SCM and configuration settings of the IoT SCM directly from within the LAN. The IoT SE as defined in this PP is not necessarily aware of the IoT device user (it does not necessarily have to know such user role), nevertheless indirectly the IoT device user is using the IoT SE (by the actions of the IoT device user with the IoT device or the IoT cloud, operations of the IoT SE – a like cryptographic operations using keys stored in the IoT SE – will be triggered). Nevertheless, the ST writer may decide to introduce the IoT device user as a role for the IoT SE and to allow the IoT device user to use authenticated services of the IoT SE, as long as no security objectives of this PP are violated.
SE developer	Developer of the IoT SE. Can generate firmware update images for the IoT SE and is the only entity that has got the keys to encrypt and sign or MAC-protect those firmware update images, if any.
SCM developer	Developer of the IoT SCM. Can generate firmware update images for the IoT SCM and is the only entity that has got the keys to encrypt and sign or MAC-protect those firmware update images.

Table 1: Terms

Asset	Description	Protection needs
IoT device data	Any data sent from the IoT device to an external network device (e.g., IoT gateway or network device belongin to IoT cloud or IoT device admin). IoT device data may originate from the IoT application, IoT SCM or IoT SE itself. Examples of IoT device data are general status data, current configuration data, consumption/billing information, etc. (the exact specification of those data cannot be given here since it depends on the concrete use case of the IoT device that utilizes the TOE). The TOE cryptographically protects authenticity and confidentiality of IoT device data before these are transmitted from the IoT device to the external network device via the IoT SCM).	Integrity/ authenticity, confidentiality
IoT admin data	Any data originating from the IoT admin, which are sent to the IoT device. Examples of IoT admin data are any kind of control data and new/updated configuration data for all parts of the IoT device, i.e. IoT application, IoT SCM or IoT SE (the exact specification of those data cannot be given here since it depends on the concrete use case of the IoT device that utilizes the TOE). The TOE cryptographically verifies authenticity and decrypts IoT admin data when these are received via the IoT SCM.	Integrity/ authenticity, confidentiality
SE ID	Identity of the SE, e.g. a unique ID or serial number for each copy of the TOE. The SE ID is stored in each copy of the TOE once (in production or personalization stage of the TOE) and never changed during the life-cycle (and therefore it can be used as an unambiguous identifier of the IoT SCM / IoT application / IoT device the TOE is integrated in and bound to).	Integrity
SE authentication key (SAK) SE message authentication key (SMK) SE confidentiality key (SCK)	Cryptographic keys that are used by the TOE to authenticate itself, outgoing IoT device data, and to decrypt incoming IoT admin data, respectively. Instead of using the keys directly, they may also serve as key derivation keys, and the correspondingly derived keys would be then used in signature generation, MAC generation and/or decryption operations. SAK shall be a static key, device-individual for each copy of the TOE. SAK cannot be output from the IoT SE. SEK and SCK may be static keys, device-individual for each copy of the TOE, or session keys established after successful mutual authentication of the TOE and the IoT device admin. The initial value for SAK and also for SMK and SCK (if the latter two are static keys) is established in each copy of the TOE before it leaves production. The successfully authenticated IoT device admin can manage these keys.	Integrity/ authenticity, confidentiality

Asset	Description	Protection needs
Admin authentication key (AAK), Admin message authentication key (AMK) Admin confidentiality key (ACK)	Cryptographic keys that are used by the TOE to authenticate the IoT device admin, incoming IoT admin data, and to encrypt outgoing IoT device data, respectively. Instead of using the keys directly, they may also serve as key derivation keys, and the correspondingly derived keys would be then used in signature verification, MAC verification and/or encryption operations. AMK and ACK may be static keys or session keys established after successful mutual authentication of the TOE and the IoT device admin. The initial value for AAK and also for AMK and ACK (if the latter two are static keys) is established in each copy of the TOE before it leaves production. The successfully authenticated IoT device admin can manage these keys.	Integrity/ authenticity, confidentiality
SE FW	All firmware parts as stored in the TOE (making up the main part of the TSF).	Integrity, confidentiality
Entropy source output, and DRG seed and state, if any	The random bits the entropy source produces and – if the TOE implements a DRG (deterministic random number generator) – the DRG seed and DRG internal state. Those values must be protected from disclosure and from any modification that they would be – in whole or part – predictable or reproducable by an attacker.	Integrity, confidentiality

Table 2: Assets

3.2 Assumptions

A.SE.Admin

It is assumed that the IoT device admin is trustworthy and well-trained to perform their duties.

A.SE.Integration

It is assumed that the IoT device manufacturer makes sure that the IoT SE TOE is integrated into the IoT host device in a way that without physical modifications of some part(s) of the IoT device the IoT SE TOE can only be used in connection with its intended IoT host device and IoT SCM. Therefore, the TOE is physically bound to the IoT host device and IoT SCM in a way that it is not easily possible to break that binding or physically inject data or commands between those parts of the IoT device. It is further assumed that the binding measure allows the IoT device manufacturer to detect if the binding has been physically tampered with (which could lead to loss of warranty or could be used as evidence in case of fraud).[1]

A.SE.Keys

It is assumed that SAK and also SMK and SCK, if these are also static keys[2], are chosen uniquely for each copy of the TOE and that each of them independently is either

1) the private part of an asymmetric key (private key of the IoT SE),
2) a symmetric key (secret key) that is randomly generated, or
3) a symmetric key (secret key) derived using a key derivation key and the SE ID (the key derivation key only known to the IoT device admin and not being stored in the TOE).

It is also assumed that all SAKs, SMKs and SCKs are pairwise different, between each other and even among different copies of the TOE. Private or secret SAKs, SMKs and SCKs generated or kept outside the TOE as well as corresponding key derivation keys, if applicable, are kept confidential at all times during production and usage of the TOE.

It is assumed that each copy of the TOE is storing the necessary values of AAK and also AMK and ACK, if these are also static keys[3], that each of them independently is either

1) the public part of an asymmetric key (public key of the IoT admin),
2) a symmetric key (secret key) that is randomly generated, or
3) a symmetric key (secret key) derived using a key derivation key and the SE ID (the key derivation key only known to the IoT device admin and not being stored in the TOE).

It is also assumed that all private or secret AAKs, AMKs and ACKs generated or kept outside the TOE as well as corresponding key derivation keys, if applicable, are kept confidential at all times during production and usage of the TOE.

Furthermore, it is assumed that SE ID is also chosen uniquely for each copy of the TOE.

[1] Strengths of binding and tamper evidence have to be decided by the IoT device manufacturer, as they typically would be interested in that the binding between IoT SCM, its IoT host device and IoT SE cannot be easily broken.

[2] If SMK and/or SCK should be session key(s), their uniqueness (per session) is not related to the OE, but subject to TOE functionality to be evaluated and vulnerability-analyzed.

[3] If SMK and/or SCK should be session key(s), their uniqueness (per session) is not related to the OE, but subject to TOE functionality to be evaluated and vulnerability-analyzed.

3.3 Threats

T.SE.Impersonation

An attacker may try to send data to the IoT device the IoT SE TOE is integrated in, impersonating the IoT device admin, or to send data to the IoT device admin, impersonating the TOE, without the respective receiving party being able to detect that. I.e. an attacker may try to fake IoT admin data or IoT device data.

The core of the attack is to trick the IoT device admin into believing that data are sent from the TOE, or to trick the TOE into believing that data are sent from the IoT device admin. Thereby, the attack may require faking the TOE's identity and/or keys stored in the TOE. Another aspect would be a man-in-the middle attack, in which an attacker could try to act between the TOE and the IoT device admin, presenting themselves as being the respective other party to TOE and the IoT device admin.

The attacker does not necessarily need access to the TOE to perform the attack, but may find other ways. They may even be an IoT device user of the IoT device the TOE is integrated in, or IoT device user of another IoT device.

T.SE.Modification

An attacker may try to intercept communication between the IoT device the IoT SE TOE is integrated in and the IoT device admin to modify or replay transmitted IoT device data or IoT admin data, without the respective receiving party being able to detect that.

The attacker has access to data sent or received by the IoT device the TOE is integrated in by eavesdropping from a network and may modify, combine or replay those data in any way (maybe also using recorded communication data from a different IoT device).

The attacker may even be a rightful IoT device user of the IoT device the TOE is integrated in, or IoT device user of a different IoT device.

T.SE.Disclosure

An attacker may try to intercept communication exchanged between the IoT device (with the IoT SE TOE inside) and the IoT device admin, to gain knowledge about transmitted IoT device data or IoT admin data.

The attacker has access to data sent or received by the IoT device the TOE is integrated in and retrieves confidential assets from that data.

T.SE.IllegalKeyAccess

An attacker may try to read out or to modify static SAK, SMK, SCK, AAK, AMK and/or ACK stored inside IoT SE TOE by logical means. An attacker may try to use any keys stored in the TOE for cryptographic operations these are not intended for according to their key type.

The attack requires access to the logical interfaces of the TOE.

The attacker may even be a rightful IoT device user of the IoT device the TOE is integrated in (trying to access data they are not authorized for), or IoT device user of a different IoT device.

3.4 Organizational security policies

OSP.SE.Auditability

The TOE shall provide functionality to output its SE firmware version on request of the IoT SCM (this request as well as the corresponding answer may be non-authenticated).

OSP.SE.PhysProt

Countermeasures against disclosing or modifying cryptographic keys stored in the TOE by tampering with the corresponding hardware shall be employed. This includes countermeasures against physical probing, physical modification, information-leakage analysis and fault-injection techniques. The countermeasures shall be suitable to protect the cryptographic keys in the TOE also against the legitimate IoT device user of the IoT device the TOE is integrated in.

OSP.SE.StrongRNG

The TOE shall provide a cryptographically strong random number generator suitable for any kind of application including generation of challenge/nonce values, symmetric keys, prime candidates (e.g., for RSA), and up to the generation of ephemeral keys for DSA or ECDSA, based on a TOE-internal entropy source (physical noise source). The random number generator including the corresponding entropy collection shall provide a security level that is consistent with all keys generated by the TOE, but no less than 100 bit.[4]

OSP.SE.StrongCrypto

All cryptographic functions used by the security functionality of the TOE shall provide a cryptographic strength of at least 100 bit.

[4] The rating of the entropy and the cryptographic strength of the generated random numbers is up to the scheme performing the TOE‘s certification (e.g., in the German CC scheme AIS20 and AIS31 would be applied).

4 Security objectives

4.1 Security objectives for the TOE

O.SE.AuthProt

The TOE shall provide functionality of data authenticity protection by adding electronic signatures or message authentication codes (MACs) to data to be sent to the IoT device admin, and by verification of electronic signatures or message authentication codes (MACs) of data received from the IoT device admin. In case such verification fails, the corresponding potentially non-authentic or corrupted data shall not be output or used TOE-internally. The authenticity-protection mechanism(s) used shall also counter undetectable replay of data and provide a security level of at least 100 bit.

O.SE.ConfProt

The TOE shall provide functionality of data confidentiality protection by encryption of IoT device data sent to an external network device or to the IoT device admin, and by decryption of encrypted IoT admin data. The encryption mechanism(s) used shall provide a security level of at least 100 bit.

O.SE.KeyAccess

The TOE shall not allow disclosure of static SAK, SMK, SCK, AAK, AMK, ACK and IDK by logical means, whereas SMK, SCK, AMK and/or ACK that are session keys generated or derived by the TOE may be read out in plaintext form from the TOE. The TOE shall restrict entering and updating of static SAK, SMK, SCK, AAK, AMK and ACK by logical means to the IoT device admin. The TOE shall not allow any keys being used in cryptographic operations they are not intended for according to their key type.

O.SE.Auditability

The TOE shall provide functionality to output its SE firmware version on request of the IoT SCM (this request may be non-authenticated).

O.SE.PhysProt

The TOE shall protect all its assets stored internally from disclosure and undetectable modification, substitution and/or deletion by physical means, including physical probing or modification, side-channel based information-leakage analysis, and fault-injection methods.

O.SE.StrongRNG

The TOE shall provide a cryptographically strong random number generator suitable for any kind of application including generation of challenge/nonce values, symmetric keys, prime candidates (e.g., for RSA), and ephemeral keys (e.g., for DSA or ECDSA or some DH or ECDH key agreement scheme using ephemeral keys), based on a TOE-internal entropy source (physical noise source). The random number generator including the corresponding entropy collection shall provide a cryptographic strength that is consistent with all keys generated by the TOE, but no less than 100 bit.[5]

[5] The rating of the entropy and the cryptographic strength of the generated random numbers is up to the scheme performing the TOE‘s certification (e.g., in the German CC scheme AIS20 and AIS31 would be applied).

4.2 Security objectives for the operational environment

OE.SE.Admin

The IoT device admin shall be trustworthy and well-trained to perform their duties.

OE.SE.Integration

The IoT device manufacturer shall make sure that the IoT SE TOE is integrated into the IoT host device in a way that without physical modifications of some part(s) of the IoT device the IoT SE TOE can only be used in connection with its intended IoT host device and IoT SCM. Therefore, the TOE shall be physically bound to the IoT host device and IoT SCM in a way that it is not easily possible to break that binding or physically inject data or commands between those parts of the IoT device. Furthermore, the binding measure shall allow the IoT device manufacturer to detect if the binding has been physically tampered with.[6]

OE.SE.Keys

The IoT device admin shall make sure that SAK and also SMK and SCK, if these are also static keys[7], are chosen uniquely for each copy of the TOE. SAK, SMK and SCK independently may be either

1) the private part of an asymmetric key (private key),
2) a symmetric key (secret key) that is randomly generated, or
3) a symmetric key (secret key) derived using a key derivation key and the SE ID (the key derivation key only known to the IoT device admin and not being stored in the TOE).

The IoT device admin shall make sure that all SAKs, SMKs and SCKs, are pairwise different, between each other and even among different copies of the TOE. Private or secret SAKs, SMKs and SCKs generated or kept outside the TOE as well as corresponding key derivation keys, if applicable, are kept confidential at all times during production and usage of the TOE.

[6] Strengths of binding and tamper evidence have to be decided by the IoT device manufacturer, as they typically would be interested in that the binding between IoT SCM, its IoT host device and IoT SE cannot be easily broken.

[7] If SMK and/or SCK should be session key(s), their uniqueness (per session) is not related to the OE, but subject to TOE functionality to be evaluated and vulnerability-analyzed.

4.3 Security objectives rationale

Security objectives / Threats, OSPs and Assumptions from SPD	O.SE.AuthProt	O.SE.ConfProt	O.SE.KeyAccess	O.SE.Auditability	O.SE.PhysProt	O.SE.StrongRNG	OE.SE.Admin	OE.SE.Integration	OE.SE.Keys
T.SE.Impersonation	X								
T.SE.Modification	X								
T.SE.Disclosure		X							
T.SE.IllegalKeyAccess			X						
OSP.SE.Auditability				X					
OSP.SE.PhysProt					X				
OSP.SE.StrongRNG						X			
OSP.SE.StrongCrypto	X	X							
A.SE.Admin							X		
A.SE.Integration								X	
A.SE.Keys									X

Table 3: Coverage of SPD items by the security objectives

T.SE.Impersonation is directly countered by **O.SE.AuthProt**, which states that the TOE shall provide authenticity protection of data exchanged with the IoT device admin using an authenticity-protection mechanism.

T.SE.Modification is directly countered by **O.SE.AuthProt**, which states that the TOE shall provide authenticity protection of data exchanged with the IoT device admin using an authenticity-protection mechanism.

T.SE.Disclosure is directly countered by **O.SE.ConfProt**, which states that the TOE shall provide confidentiality protection of data exchanged with the IoT device admin by encryption.

T.SE.IllegalKeyAccess is directly countered by **O.SE.KeyAccess**, which states that the TOE shall limit logical access to its keys concerning entry and update to the IoT device admin, and that keys shall only be usable for their intended cryptographic operation.

OSP.SE.Auditability is directly enforced by **O.SE.Auditability** (objective re-states OSP).

OSP.SE.PhysProt is directly enforced by **O.SE.PhysProt** (objective re-states OSP).

OSP.SE.StrongRNG is directly enforced by **O.SE.StrongRNG** (objective re-states OSP).

Concerning cryptographic functions used for authenticity-protection and encryption of data exchanged with the IoT admin, **OSP.SE.StrongCrypto** is enforced by the combination of **O.SE.AuthProt** and **O.SE.ConfProt**, which state that the authenticity-protection mechanism and the encryption shall have a security level of at least 100 bit.

A.SE.Admin is directly upheld by **OE.SE.Admin** (objective re-states assumption).

A.SE.Integration is directly upheld by **OE.SE.Integration** (objective re-states assumption).

A.SE.Keys is directly upheld by **OE.SE.Keys** (objective re-states assumption).

5 Extended components definition

5.1 Definition of the family generation of random numbers (FCS_RNG)

Family behaviour

This section describes the functional requirements for the generation of random numbers, which may be used as secrets for cryptographic purposes or authentication. The IT security functional requirements for a TOE are defined in an additional family (FCS_RNG) of the Class FCS (Cryptographic support). The requirements address the type of the random number generator as defined in AIS 20 and AIS 31 and quality of the random numbers.

Component levelling

FCS_RNG.1, Generation of random numbers, requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

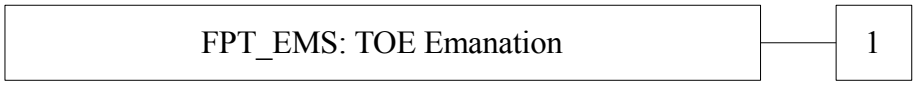
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	**The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator that implements: [assignment: list of security capabilities].**
FCS_RNG.1.2	**The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].**

5.2 Definition of the family TOE emanation (FPT_EMS)

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling



Management: FPT_EMS.1

There are no management activities foreseen.

Audit: FPT_EMS.1

There are no actions defined to be auditable.

FPT_EMS.1 TOE emanation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 **The TOE shall not emit [assignment: *types of emissions*] in excess of [assignment: *specified limits*] enabling access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].**

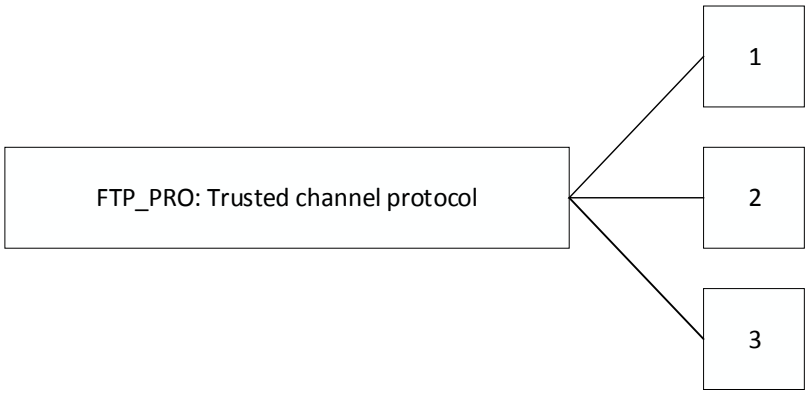
FPT_EMS.1.2 **The TSF shall ensure [assignment: *type of users*] are unable to use [assignment: *types of interfaces/ports*] to gain access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].**

5.3 Definition of the family trusted channel protocol (FTP_PRO)

Family behaviour

This family defines requirements for establishing a trusted channel and using the trusted channel to transfer the TSF data or user data securely.

Component levelling



FTP_PRO.1 Trusted channel protocol requires that communication be established in accordance with a defined protocol.

FTP_PRO.2 Trusted channel establishment requires that keys be securely established between the peers.

FTP_PRO.3 Trusted channel data protection requires that data in transit be protected.

Management of FTP_PRO.1

The following actions could be considered for the management functions in FMT:

a) Configuring the protocols needed for the trusted channel

b) Configuring the credentials for using the trusted channel

c) Configuring the conditions for initializing and terminating the trusted channel.

Management of FTP_PRO.2

The following actions could be considered for the management functions in FMT:

a) Configuring the parameters for shared secrets

b) Configuring the parameters for cryptographic key derivation.

Management of FTP_PRO.3

The following actions could be considered for the management functions in FMT:

a) Configuring the encryption and integrity mechanisms used by the trusted channel.

Audit of FTP_PRO.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failure of the trusted channel establishment

b) Minimal: Identification of the initiator and target of failed trusted channel establishment

c) Basic: All attempted uses of the trusted channel

d) Basic: Identification of the initiator and target of all trusted channel attempts.

Other events should be considered according to the specific protocols used.

Audit of FTP_PRO.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Authentication failures during channel establishment

b) Basic: All authentication attempts.

Audit of FTP_PRO.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failures when attempting to verify channel properties in FTP_PRO.3.2.

FTP_PRO.1 Trusted channel protocol

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.2 Trusted channel establishment
	FTP_PRO.3 Trusted channel data protection.
FTP_PRO.1.1	**The TSF shall implement [assignment: *trusted channel protocol*] acting as [assignment: *defined protocol role(s)*] in accordance with: [assignment: *list of standards*].**
FTP_PRO.1.2	**The TSF shall enforce usage of the trusted channel for [assignment: *purpose(s) of the trusted channel*] in accordance with: [assignment: *list of standards*].**
FTP_PRO.1.3	**The TSF shall permit [selection: *itself, its peer*] to initiate communication via the trusted channel.**
FTP_PRO.1.4	**The TSF shall enforce the following rules for the trusted channel: [assignment: *rules governing operation and use of the trusted channel and/or its protocol*].**
FTP_PRO.1.5	**The TSF shall enforce the following static protocol options: [assignment: *list of options and references to standards in which each is defined*].**
FTP_PRO.1.6	**The TSF shall negotiate one of the following protocol configurations with its peer: [assignment: *list of configurations and reference to standards in which each is defined*].**

User application notes

FTP_PRO.1 may be iterated by the PP/ST author for different protocols, but also for different protocol roles of the same protocol, if completion of FTP_PRO.1 operations needs to be different for each protocol role.

Where values used in the completion of FTP_PRO.1 operations have dependencies between different FTP_PRO.1 elements, these need to be made clear in the instantiation of FTP_PRO.1. For example, a table could be given in which the columns represent the relevant selections and assignments, and the rows define the valid combination of completion values.

Operations

Assignment:

In FTP_PRO.1, examples of “defined protocol roles” would be ‘client’ or ‘server’ (e.g. in case of TLS protocol), ‘initiator’ or ‘responder’ (e.g., in case of IKEv2/IPsec protocol), ‘Trust Center’ (e.g., in case of ZigBee protocol) or ‘Key Distribution Centre’ (e.g., in case of Kerberos protocol).

In the first assignment in FTP_PRO.1.5, the PP/ST author should state rules for when the secure channel is required to be used by the TOE, such as mandating its use for communications with an audit server. If no specific uses of the channel are mandated for the TOE, this assignment can be completed with “none specified” (in this case, also the second assignment shall be completed with “none specified”).

In FTP_PRO.1.5, the PP/ST author should state rules related to implementation of the protocol (e.g., rules on maximum packet sizes or rekeying intervals). If there are no rules required, or if the standards referenced in other elements of FTP_PRO.1 include the relevant rules and no specific evaluator check is required for the context in which FTP_PRO.1 is being used, this assignment can be completed with “none specified”.

In FTP_PRO.1.6, the PP/ST author should state rules related to negotiable aspects of the protocol, when intending to narrow the options provided by the TOE compared to the standard that defines the protocol (e.g., selection of cipher suites or acceptance of older protocol versions). If no rules are required, this assignment can be completed with “none specified”. Where the assignment is completed with a list then that list specifies the only configurations permitted – any other configuration would be a violation of the SFR. FTP_PRO.1.6 may be used to specify mandatory supported configurations without limiting the TOE to using these configurations by, for example, listing the required configurations with “(support required)” after each entry in the list and then including a final element which states that any other configuration permitted by the standard is allowed.

FTP_PRO.2 Trusted channel establishment

Hierarchical to: No other components.

Dependencies: FTP_PRO.1 Trusted channel protocol

[FCS_CKM.1 Cryptographic key generation, or
FCS_CKM.2 Cryptographic key distribution]

FCS_CKM.5 Cryptographic key derivation

FCS_COP.1 Cryptographic operation.

FTP_PRO.2.1 **The TSF shall establish a shared secret with its peer using one of the following mechanisms: [assignment: *list of key establishment mechanisms*].**

FTP_PRO.2.2 **The TSF shall authenticate [selection*: its peer, itself to its peer*] using one of the following mechanisms: [assignment: *list of authentication mechanisms*] and according to the following rules: [assignment: *list of rules for carrying out the authentication*].**

FTP_PRO.2.3 **The TSF shall use [assignment: *key derivation function*] to derive the following cryptographic keys from a shared secret: [assignment: *list of cryptographic keys*].**

User application notes

For each iteration of FTP_PRO.1 by the PP/ST author, which represents a different protocol, a corresponding iteration of FTP_PRO.2 is needed in the PP/ST. For iterations of FTP_PRO.1 by the PP/ST author, which only express the behaviour of the TSF for different protocol roles of the same protocol, the same instantiation of FTP_PRO.2 may be suitable to fulfil the dependency of such FTP_PRO.1 iterations.

Operations

Assignment:

In FTP_PRO.2.2, the PP/ST author may use the 'list of rules for carrying out the authentication' to limit available parameters for the authentication mechanisms. For example, rules might be stated for the format (e.g. FQDN or IP address, use of wildcards) or prioritisation of identifiers when alternative sources of an identifier are available in the authentication data exchanged.

FTP_PRO.3 Trusted channel data protection

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol
	FTP_PRO.2 Trusted channel establishment
	FCS_COP.1 Cryptographic operation.

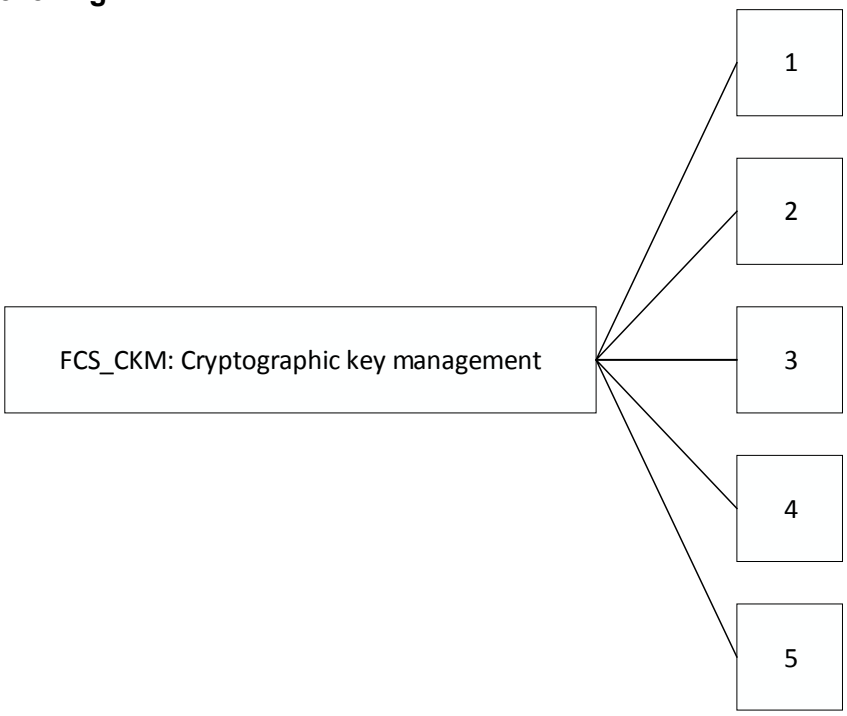
FTP_PRO.3.1 **The TSF shall protect data in transit from unauthorised disclosure using one of the following mechanisms: [assignment: *list of encryption mechanisms*].**

FTP_PRO.3.2 **The TSF shall protect data in transit from [selection: *modification, deletion, insertion, replay,* [assignment: *other*]] using one of the following mechanisms: [assignment: *list of integrity protection mechanisms*].**

5.4 Definition of the component cryptographic key derivation (FCS_CKM.5)

This chapter describes functional requirements for key derivation as process by which one or more keys are calculated from either a pre-shared key or a shared secret and other information. The component is part of the family FCS_CKM of the class FCS. The component FCS_CKM.5 has been specified as follows:

Component levelling



Management: FCS_CKM.5

There are no management activities foreseen.

Audit: FCS_CKM.5

There are no actions defined to be auditable.

FCS_CKM.5 Cryptographic key derivation

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
	FCS_CKM.4 Cryptographic key destruction

FCS_CKM.5.1 **The TSF shall derive cryptographic keys [assignment: *key type*] from [assignment: *input parameters*] in accordance with a specified cryptographic key derivation algorithm [assignment: *cryptographic key derivation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

6 Security requirements

6.1 Security functional requirements

In the following subsections the security functional requirements for the IoT SE TOE are stated, grouped according to the functionality they are belonging to. For all operations in the SFRs, which have been at least partly executed in comparison with the original CC definition, the original version of the operation text as defined in CC part 2 are given in form of end notes in this PP (search for the corresponding number in Roman numerals in section 8.3).

6.1.1 Trusted path

FTP_PRO.1 Trusted channel protocol

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.2 Trusted channel key establishment
	FTP_PRO.3 Trusted channel data protection
FTP_PRO.1.1	**The TSF shall implement [assignment: *trusted channel protocol*] acting as [assignment: *defined protocol role(s)*] in accordance with: [assignment: *list of standards*].**
FTP_PRO.1.2	**The TSF shall enforce usage of the trusted channel for [assignment: *purpose(s) of the trusted channel*] in accordance with: [assignment: *list of standards*].**
FTP_PRO.1.3	**The TSF shall permit [selection: *itself, its peer*] to initiate communication via the trusted channel.**
FTP_PRO.1.4	**The TSF shall enforce the following rules for the trusted channel: [assignment: *rules governing operation and use of the trusted channel and/or its protocol*].**
FTP_PRO.1.5	**The TSF shall enforce the following static protocol options: [assignment: *list of options and references to standards in which each is defined*].**
FTP_PRO.1.6	**The TSF shall negotiate one of the following protocol configurations with its peer: [assignment: *list of configurations and reference to standards in which each is defined*].**
AN(FTP_PRO.1):	The ST/PP author shall model both, trusted channel between the TSF and a network device and trusted path (i.e. end-to-end secured connection) between the TSF and the IoT device admin, by FTP_PRO.1. If different protocols are used to realize trusted channel and trusted path, the ST/PP author shall iterate FTP_PRO.1 as FTP_PRO.1/TC and FTP_PRO.1/TP. Furthermore, according to the user application notes for FTP_PRO.1, the ST/PP author may have to further iterate FTP_PRO.1 (or FTP_PRO.1/TC and/or FTP_PRO.1/TP, if applicable) for different protocol roles.

FTP_PRO.2 Trusted channel establishment

Hierarchical to: No other components.

Dependencies: FTP_PRO.1 Trusted channel protocol

[FCS_CKM.1 Cryptographic key generation, or
FCS_CKM.2 Cryptographic key distribution]

FCS_CKM.5 Cryptographic key derivation

FCS_COP.1 Cryptographic operation

FTP_PRO.2.1 **The TSF shall establish a shared secret with its peer using one of the following mechanisms: [assignment: *list of key establishment mechanisms*].**

FTP_PRO.2.2 **The TSF shall authenticate [selection: *its peer, itself to its peer*] using one of the following mechanisms: [assignment: *list of authentication mechanisms*] and according to the following rules: [assignment: *list of rules for carrying out the authentication*].**

FTP_PRO.2.3 **The TSF shall use [assignment: *key derivation function*] to derive the following cryptographic keys from a shared secret: [assignment: *list of cryptographic keys*].**

AN(FTP_PRO.2): The ST/PP author shall model both, trusted channel establishment between the TSF and a network device and trusted path (i.e. end-to-end secured connection) establishment between the TSF and the IoT device admin, by FTP_PRO.2. If different protocols are used to realize trusted channel and trusted path, the ST/PP author shall iterate FTP_PRO.2 as FTP_PRO.2/TC and FTP_PRO.2/TP.
To satisfy remaining open dependencies of FTP_PRO.2, the ST/PP author has to include FCS_CKM.1 or FCS_CKM.2 in the ST/PP according to the actual key management related to the chosen trusted channel protocols.

FTP_PRO.3 Trusted channel data protection

Hierarchical to: No other components.

Dependencies: FTP_PRO.1 Trusted channel protocol

FTP_PRO.2 Trusted channel key establishment

FCS_COP.1 Cryptographic operation

FTP_PRO.3.1 **The TSF shall protect data in transit from unauthorised disclosure using one of the following mechanisms: [assignment: *list of encryption mechanisms*].**

FTP_PRO.3.2 **The TSF shall protect data in transit from [selection: *modification, deletion, insertion, replay, [assignment: other]*] using one of the following mechanisms: [assignment: *list of integrity protection mechanisms*].**

AN(FTP_PRO.3): The ST/PP author shall model both, trusted channel data protection between the TSF and a network device and trusted path (i.e. end-to-end) data protection between the TSF and the IoT device admin, by FTP_PRO.3. If different protocols are used to realize trusted channel and trusted path, the ST/PP author shall iterate FTP_PRO.3 as FTP_PRO.3/TC and FTP_PRO.3/TP.

FCS_CKM.5 Cryptographic key derivation

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.5.1 **The TSF shall derive cryptographic keys [assignment: *key type*] from [assignment: *input parameters*] in accordance with a specified cryptographic key derivation algorithm [assignment: *cryptographic key derivation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

AN(FCS_CKM.5): The ST/PP author shall iterate FCS_CKM.5 if necessary to cover all corresponding dependencies concerning cryptographic key derivation arising from FTP_PRO.2 or iterations thereof.

According to the dependencies of FCS_CKM.5, the ST/PP author shall further include the necessary FCS_CKM.2, FCS_COP.1 and/or FCS_CKM.4 components, to cover all corresponding cryptographic key derivation mechanisms as specified in FCS_CKM.5 or iterations thereof.

6.1.2 SE key access

FDP_ACC.1/SEkey Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/SEkey **The TSF shall enforce the *IoT SE key access policy*[i] on**
(1) objects: SAK, SMK, SCK, AAK, AMK, ACK;
***(2) operations: key update, session key generation/derivation, key output, signature/MAC generation, signature/MAC verification, encryption, decryption*[ii].**

FDP_ACF.1/SEkey Security attribute based access control

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/SEkey **The TSF shall enforce the *IoT SE key access policy*[iii] to objects based on the following:**
(1) objects: SAK, SMK, SCK, AAK, AMK, ACK;
***(2) attributes: key type (private, public, secret, session), key usage type (authentication, confidentiality), admin signature/MAC][iv].**

FDP_ACF.1.2/SEkey **The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:**
(1) key update is only allowed for a key if the admin signature/MAC over the key update request including the key data is successfully verified][v];
(2) session key generation/derivation is only allowed for a key with the key type session.

FDP_ACF.1.3/SEkey **The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:**
(1) key output is always allowed for a key with key type public or session;

(2) signature/MAC generation is always allowed for a key with usage type authentication and key type private or secret;
(3) signature/MAC verification is always allowed for a key with usage type authentication and key type public or secret;
(4) encryption is always allowed for a key with usage type confidentiality and key type public or secret;
(5) encryption is always allowed for a key with usage type confidentiality and key type private or secret[vi].

FDP_ACF.1.4/SEkey **The TSF shall explicitly deny access of subjects to objects based on the following additional rules:**
(1) key output is never allowed for a key with key type private or secret;
(2) signature/MAC generation is never allowed for a key with usage type confidentiality or key type public;
(3) signature/MAC verification is never allowed for a key with usage type confidentiality or key type private;
(4) encryption is never allowed for a key with usage type authentication;
(5) decryption is never allowed for a key with usage type authentication[vii].

AN(FDP_ACF.1/SEkey) The dependency to FMT_MSA.3 is not applicable. There are no default values for the attributes of this access control policy.

6.1.3 TOE management

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 **The TSF shall maintain the roles ~~*IoT device admin*~~[viii].**

FMT_SMR.1.2 **The TSF shall be able to associate users with roles.**

FIA_UID.1 Timing of identification

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

FIA_UID.1.1 **The TSF shall allow *querying version information of the TOE*[ix] on behalf of the user to be performed before the user is identified.**

FIA_UID.1.2 **The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.**

AN(FIA_UID.1): The IoT device admin is identified and authenticated during establishment of a trusted path between the TSF and the IoT device admin, therefore there is no need for the TOE developer to come up with an additional identification and authentication mechanism for the IoT device admin.

FIA_UAU.1 Timing of authentication

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 **The TSF shall allow *querying version information of the TOE*[x] on behalf of the user to be performed before the user is authenticated.**

FIA_UAU.1.2 **The TSF shall require each user to be successfully authenticated before**

AN(FIA_UAU.2): The IoT device admin is identified and authenticated during establishment of a trusted path between the TSF and the IoT device admin, therefore there is no need for the TOE developer to come up with an additional identification and authentication mechanism for the IoT device admin.

FMT_SMF.1 Specification of management functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 **The TSF shall be capable of performing the following management functions: *query version information of the TOE*[xi].**

6.1.4 Physical protection

FPT_PHP.3 Resistance to physical attack

Dependencies: None.

FPT_PHP.3.1 **The TSF shall resist physical probing, physical manipulation and fault injection with the objective to disclose or modify cryptographic keys or to modify TSF data in the TOE[xii] to the TSF elements storing or processing cryptographic keys or TSF data[xiii] by responding automatically such that the SFRs are always enforced.**

FPT_EMS.1 TOE emanation

Hierarchical to: No other components.

Dependencies: None.

FPT_EMS.1.1 **The TOE shall not emit information in terms of electromagnetic emanation, power consumption or timing[xiv] in excess of [assignment: specified limits] enabling access to SE firmware[xv] and cryptographic keys except session keys that are exportable from the TOE anyway[xvi].**

FPT_EMS.1.2 **The TSF shall ensure *all users*[xvii] are unable to use any kind of TOE interface/port[xviii] to gain access to *SE firmware*[xix] and *cryptographic keys except session keys that are exportable from the TOE anyway*[xx].**

6.1.5 Random number generation

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: None.

FCS_RNG.1.1 **The TSF shall provide a [selection: physical, deterministic, hybrid physical, hybrid deterministic][xxi] random number generator that implements: [assignment: list of security capabilities].**

FCS_RNG.1.2 **The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].**

Remark: RNG type “non-physical true” has been removed here compared to the SFR definition, as it would be not meaningful for an IoT SE.

AN(FCS_RNG.1): In FCS_RNG.1, the ST author has to add the requirements concerning the random number generation in coordination with the corresponding certification body and with regards to the applicable requirements, in particular concerning the security capabilities and quality metric. The ST author shall make sure that the choice of the operations has to be suitable that the random numbers output by the RNG can be used in all cryptographic functions of IoT SE itself, but also of the IoT SCM, that require a minimal security level of 100 bit.

6.1.6 Cryptographic operation

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 **The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

AN(FCS_COP.1): There are several SFRs in this PP, which model functionality making use of cryptographic operations. The author of this PP cannot decide, how many different cryptographic operations (also in terms of cryptographic algorithm, key size and applicable standard) would be necessary for a concrete TOE conformant to this PP. To avoid that this PP is bloated up with a lot of iterations of FCS_COP.1, which in the end could lead to a highly redundant set of SFRs in the ST/PP based on this PP, it is left open to the ST/PP author to iterate FCS_COP.1 in a way that all SFR dependencies requiring FCS_COP.1 are satisfied, and that also all cryptographic operations, which are needed to cover the security objectives of the TOE, are included in the final set of SFRs of the ST/PP. (Completeness of the FCS_COP.1 iterations will have to be shown in the ST/PP in terms of the SFR dependency rationale and the security objectives rationale anyway.)

Furthermore, as the dependencies concerning the key management related to the cryptographic operation modelled by FCS_COP.1, i.e. FDP_ITC.1, FDP_ITC.2, FCS_CKM.1 and FCS_CKM.4,

- may be satisfied very differently for different concrete TOEs,
- may be satisfied very differently even for different keys of the same TOE,
- may be rightfully left unsatisfied with a corresponding rationale given, or

- may be satisfied by the very same iteration of FDP_ITC.1, FDP_ITC.2, FCS_CKM.1 and/or FCS_CKM.4 even for several iterations of FCS_COP.1,

none of these dependencies SFRs have been included in this PP already. It is up to the ST/PP author to make sure that all those dependencies will be satisfied for all iterations of FCS_COP.1 as finally stated in the ST/PP. Satisfaction of dependencies has to be shown in the SFR dependency rationale in the ST/PP for all iterations of all SFRs independently anyway.

To still allow a somehow meaningful dependency rationale and security objectives rationale in this PP, in the following the dependencies and security functional requirements needing instances/iterations of FSC_COP.1 in the ST/PP are listed:

- cryptographic operation needed for FTP_PRO.2 shared secret establishment,
- cryptographic operation needed for FTP_PRO.2 key derivation,
- cryptographic operations ‘encryption and decryption’ according to FTP_PRO.3,
- cryptographic operation ‘integrity protection’ according to FTP_PRO.3.

In each iteration of FCS_COP.1 in the ST/PP, in the assignment about the ‘list of cryptographic operations’ the ST/PP author shall also enter the corresponding keys being used, e.g., ‘signature/MAC verification using SCM-FAK’ or ‘decryption using SCM-FCK’. This will allow to easier map the FCS_COP.1 iterations to the related dependencies and security objectives, respectively.

Finally, for all iterations of FCS_COP.1 the choice of cryptographic algorithms and cryptographic key sizes has to ensure the required minimum security level of 100 bit for all cryptographic operations in their corresponding use case/protocol.

Remark: The ST/PP author shall take note that in the Package “Secure Update” hereinafter already two iterations of FCS_COP.1 are included, not to get in naming conflict of FCS_COP.1 iterations in case the package is used.

6.2 Security assurance requirements

The security assurance requirements for this TOE shall be EAL4 augmented by AVA_VAN.4 as defined in CC Part 3:

Assurance class	Assurance components	
ADV: Development	ADV_ARC.1	Security architecture description
	ADV_FSP.4	Complete functional specification
	ADV_IMP.1	Implementation representation of the TSF
	ADV_TDS.3	Basic modular design
AGD: Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC: Life-cycle support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
ASE: Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
ATE: Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: basic design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.4	Methodical vulnerability analysis (augmented)

Table 4: Security assurance requirements (EAL4 augmented by AVA_VAN.4 and ALC_FLR.1)

6.3 Security requirements rationale

6.3.1 Security functional requirement (SFR) rationale

6.3.1.1 Fulfilment of the security objectives for the TOE

The following table shows that all SFRs chosen trace back to TOE security objectives:

Security objectives / Security functional requirements	O.SE.AuthProt	O.SE.ConfProt	O.SE.KeyAccess	O.SE.Auditability	O.SE.PhysProt	O.SE.StrongRNG
FTP_PRO.1	X	X				
FTP_PRO.2	X	X				
FTP_PRO.3	X	X				
FCS_CKM.5	X	X				
FCS_COP.1	X	X				
FDP_ACC.1/SEkey			X			
FDP_ACF.1/SEkey			X			
FMT_SMR.1				X		
FMT_UID.1				X		
FMT_UAU.1				X		
FMT_SMF.1				X		
FPT_PHP.3					X	
FPT_EMS.1					X	
FCS_RND.1						X

Table 5: Tracing back security requirements to TOE security objectives

The following table shows that the security functional requirements fulfil the security objectives for the TOE:

TOE security objective	SFR	Rationale
O.SE.AuthProt	FTP_PRO.1	Defines the requirement to use a well-defined trusted channel protocol including protocol options, operational rules, allowed configurations, etc. and is therefore the base for the authenticity protection from the objective
	FTP_PRO.2	Defines the requirement for well-defined authentication and key establishment mechanisms in the trusted channel protocol. Authentication directly contributes to meeting the objective, the key establishment may be used as a base to derive further data authentication keys (e.g., session keys)
	FTP_PRO.3	Defines the requirement for well-defined key derivation mechanisms in the trusted channel protocol, which may be used to derive further data authentication keys (e.g., session keys)
	FCS_CKM.5	Defines the requirement to use a specific standardized key derivation algorithm with specified key size
	FCS_COP.1	Defines the requirement to use a specific standardized cryptographic operation (primitive) as part of the key derivation algorithm
O.SE.ConfProt	FTP_PRO.1	Defines the requirement to use a well-defined trusted channel protocol including protocol options, operational rules, allowed configurations, etc. and is therefore the base for the confidentiality protection from the objective
	FTP_PRO.2	Defines the requirement for well-defined authentication and key establishment mechanisms in the trusted channel protocol. Authentication directly contributes to meeting the objective, the key establishment may be used as a base to derive further data authentication keys (e.g., session keys)
	FTP_PRO.3	Defines the requirement for well-defined key derivation mechanisms in the trusted channel protocol, which may be used to derive further data authentication keys (e.g., session keys)
	FCS_CKM.5	Defines the requirement to use a specific standardized key derivation algorithm
	FCS_COP.1	Defines the requirement to use a specific standardized cryptographic operation (primitive) as part of the key derivation algorithm
O.SE.KeyAccess	FDP_ACC.1/ SEkey	Defines the requirement for a connection control policy and defines the corresponding objects (external network devices) and operations (connection establishment)
	FDP_ACF.1/ SEkey	Defines the requirement for security-attribute based access control for the connection establishment, the corresponding security attributes and the rules allowing only those connections, which have been configure in terms of connection control rules (security attribute). Requested connections, which are not configured at all or whose connection rules do not match the request, are denied.

TOE security objective	SFR	Rationale
O.SE.Auditability	FMT_SMR.1	Defines the requirement that the TOE is aware of the necessary role
	FMT_UID.1	Defines the requirement that querying version of the TOE is possible prior to user identification
	FMT_UAU.1	Defines the requirement that querying version of the TOE and the network control rules is possible prior to user authentication
	FMT_SMF.1	Defines the requirement that functionality for querying TOE version is provided by the TOE
O.SE.PhysProt	FPT_PHP.3	Defines the requirement that cryptographic keys inside the TOE hall be protected against physical probing and manipulation, and against fault injection attacks
	FPT_EMS.1	Defines the requirement that SE firmware and cryptographic keys except session keys inside the TOE shall be protected against disclosure by electromagnetic emanation, power consumption or timing information by all users via all interfaces/ports of the TOE
O.SE.StrongRNG	FCS_RND.1	Defines the requirements for a random number generator to be implemented in the TOE, together with its characteristics and quality metrics. The RNG type “non-physical true” has been removed from the possible choice as it would not be meaningful for the TOE. (Judgement, whether after completion of the operations in the ST an RNG suitable to serve 100 bit security level is defined, is up to the CC certification body performing the certification of the particular TOE.)

Table 6: Mapping of security requirements to TOE security objectives

6.3.1.2 Fulfilment security functional requirements (SFR) dependencies

Following table shows that dependencies of the SFRs are satisfied within this PP or rationale is referenced (typically provided in SFR application notes “AN”), why a dependency is either not applicable at all or why it has been left to be satisfied by the ST/PP developer:

SFR	Dependency	Satisfied in this PP?
FTP_PRO.1	FTP_PRO.2 FTP_PRO.3	Yes Yes
FTP_PRO.2	FTP_PRO.1 FCS_CKM.1 or FCS_CKM.2 FCS_CKM.5 FCS_COP.1	Yes No, qualified by AN(FTP_PRO.2) Yes, qualified by AN(FCS_CKM.5) Yes, qualified by AN(FCS_COP.1)
FTP_PRO.3	FTP_PRO.1 FTP_PRO.2 FCS_COP.1	Yes Yes Yes, qualified by AN(FCS_COP.1)
FCS_CKM.5	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	No, qualified by AN(FCS_CKM.5) No, qualified by AN(FCS_CKM.5)
FDP_ACC.1/SEkey	FDP_ACF.1	Yes (by FDP_ACF.1/SEkey)

SFR	Dependency	Satisfied in this PP?
FDP_ACF.1/SEkey	FDP_ACC.1 FMT_MSA.3	Yes (by FDP_ACC.1/SEkey No, not applicable as qualified by AN(FDP_ACF.1/SEkey)
FMT_SMR.1	FIA_UID.1	Yes
FMT_UID.1	none	
FMT_UAU.1	FIA_UID.1	Yes
FMT_SMF.1	none	
FPT_PHP.3	none	
FPT_EMS.1	none	
FCS_RNG.1	none	
FCS_COP.1	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No, qualified by AN(FCS_COP.1) No, qualified by AN(FCS_COP.1)

Table 7: Satisfaction of SFR dependencies

6.3.2 Security assurance requirement (SAR) rationale

The primary use case for the IoT SE is to be built in IoT host devices like home appliances. Any kind of network-based attacks are deemed to be countered by the use of strong cryptographic security functionality, providing a security level of at least 100 bit. It is assumed that the IoT device is running in a household with limited physical access, still the IoT device user (typically owning or having rented/leased the IoT device) could act as an attacker, using direct physical means or side-channel attack methods. To gain reasonable assurance that the security functionality is countering that threat, evaluation assurance level **EAL4** was chosen, as it is the lowest evaluation assurance level that required evaluation on the level of the implementation representation (i.e., source code), enabling a meaningful level of vulnerability analysis.

From the home appliance use case and the corresponding financial risk related to compromising the TOE's assets, the resistance level of the TOE does not need to be very high (in contrast to e.g., payment or banking applications). On the other hand, home appliance IoT devices might be in the field for more than 10 years, therefore it is also not desired to require a too low resistance level from the very beginning. For these reasons, as a reasonable compromise between resistance level and efforts for implementation, evaluation and certification, vulnerability assessment component **AVA_VAN.4** (providing assurance concerning **resistance of the TOE against attackers possessing *moderate* attack potential**) was chosen for the TOE.

Thus, for being conformant to this PP, the ST writer shall claim EAL4 augmented with AVA_VAN.4.

Furthermore, if the TOE supports firmware update, the correspondingly required functional package "Secure Update" makes sure that the firmware update functionality and the relevant measures in the operational environment meet a particular minimal security level and are well-comparable between different TOEs certified according to this PP.

7 Package “Secure Update”

This PP does not require firmware update functionality for the IoT SE TOE, but in case the ST claiming compliance to this PP should require firmware update functionality for the IoT SE TOE, the package defined in this section shall be used to model it. The assets, assumption, OSP, security objectives and SFRs contain a SE firmware authentication key (SE-FAK) and a SE firmware confidentiality key (SE-FCK) in addition to the assets listed above. Guidance on how to define SE-FAK and SE-FCK can be found in the IoT-SCM-PP, in form of the analogous corresponding definitions of SCM-FAK and SCM-FCK assets.

7.1 Package “Secure Update” – Security problem definition

7.1.1 Package “Secure Update” – Additional assets

Asset	Description	Protection needs
SE FW update image	An authenticity-protected and confidentiality-protected firmware update image that is imported into the TOE to update/replace in whole or part the current TOE firmware. Presented to the TOE during the firmware update process, and stored/activated inside the TOE if authenticity verification and decryption is successful.	Integrity/ authenticity, confidentiality
SE FW update version	Attribute of the SE FW update image specifying its version. Presented to the TOE during the firmware update process, and stored as latest SE FW version in the TOE if the update is successful.	Integrity/ authenticity
Latest SE FW version	Attribute of the last successfully installed firmware update, specifiying its version. TSF data, which is stored persistently in the TOE.	Integrity
SE FW authentication key (SE-FAK)	Public key or secret key used to verify the authenticity of a presented SE FW update image, randomly generated by the IoT SE developer. SE-FAK can be updated in the TOE (using the same authenticity-protection and confidentiality-protection mechanisms used for the SE FW update image). If SE-FAK is a secret key, it shall be device-individual for each copy of the TOE.	Integrity/ authenticity, if secret key also confidentiality
SE FW confidentiality key (SE-FCK)	Private key or secret key used to decrypt a presented SE FW update image, randomly generated by the IoT SE developer. SE-FCK can be updated in the TOE (using the same authenticity-protection and confidentiality-protection mechanisms used for the SE FW update image).	Integrity/ authenticity, confidentiality

Asset	Description	Protection needs
SE-FAK signature/MAC	During firmware update: attribute of the SE FW update image and its version, in terms of a signature or MAC over both. Presented to the TOE during the firmware update process. Can only be generated by the IoT SE developer, as only them shall know the necessary private key or as only them shall have the MAC key as stored in the TOE, respectively. During update of SE-FAK and/or SE-FCK: attribute of the value of the SE-FAK and/or SE-FCK to be updated, in terms of a signature or MAC over the value(s), which is verified by the TOE (using the currently stored SE-FAK). SE-FAK signature/MAC is presented to the TOE during the firmware key update process. SE-FAK signature/MAC can only be generated by the IoT SE developer, as only them shall know the necessary private key or as only them shall have the MAC key as stored in the TOE, respectively.	None (provides integrity/ authenticity protection itself)

Table 8: Additional assets for package "Secure Update"

7.1.2 Package "Secure Update" – Additional assumptions

A.SE.FirmwareKeys

If SE-FAK is a public key (for verification of a signature), it is assumed that the IoT SE developer generates a corresponding key pair randomly and keeps the corresponding private key confidentiality-protected in their development environment. It is further assumed that a public SE-FAK is only shared for firmware updates for those IoT SE products, which can install/execute identical SE FW Update Images; whereas for IoT SE products which cannot, different product-specific public SE-FAKs are used by the IoT SE developer.

If SE-FAK is a secret key (for verification of a MAC), it is assumed that the IoT SE developer chooses it device-individual, either by random generation or by key derivation, and that the IoT SE developer keeps SE-FAK and its related key derivation key (if any) confidentiality-protected in their development environment. A key derivation key is only shared for deriving SE-FAK for firmware updates for those IoT SE products, which can install/execute identical SE FW Update Images; for IoT SE products which cannot, different product-specific key derivation keys for derivation of SE-FAKs are used by the IoT SE developer.

7.1.3 Package "Secure Update" – Additional threats

None.

7.1.4 Package "Secure Update" – Additional organizational security policies

OSP.SE.SecureUpdate

The TOE shall provide functionality to securely update its firmware or parts thereof, protected concerning authenticity and confidentiality. Only authentic SE firmware update images as provided by the developer of the TOE shall be accepted by the TOE. Non-authentic SE firmware update images or those being issued by the TOE developer, but modified thereafter shall be rejected by the TOE. The TOE shall not accept a SE firmware update image, if its firmware version is older than the version of the latest successfully installed firmware. The keys to protect the authenticity and confidentiality of the SE firmware update image, i.e. SE-

FAK and SE-FCK, respectively, shall be updateable, this update protected concerning authenticity and confidentiality the same way as the SE firmware update image itself. The authenticity-protection mechanism and the confidentiality-protection mechanism used shall provide a cryptographic security level of at least 100 bit.

7.2 Package “Secure Update” – Additional security objectives

7.2.1 Package “Secure Update” – Additional security objectives for the TOE

O.SE.SecureUpdate

The TOE shall provide functionality to securely update its firmware or parts thereof, protected concerning authenticity and confidentiality. Only authentic SE firmware update images as provided by the developer of the TOE shall be accepted by the TOE. Non-authentic SE firmware update images or those being issued by the TOE developer, but modified thereafter shall be rejected by the TOE. The TOE shall not accept a SE firmware update image, if its firmware version is older than the version of the latest successfully installed firmware. The keys to protect the authenticity and confidentiality of the SE firmware update image, i.e. SE-FAK and SE-FCK, respectively, shall be updateable, this update protected concerning authenticity and confidentiality the same way as the SE firmware update image itself. The authenticity-protection mechanism and the confidentiality-protection mechanism used shall provide a cryptographic security level of at least 100 bit.

7.2.2 Package “Secure Update” – Additional security objectives for the operational environment

OE.SE.FirmwareKeys

If SE-FAK is a public key (for verification of a signature), the IoT SE developer shall generate a corresponding key pair randomly and keep the corresponding private key confidentiality-protected in their development environment. A public SE-FAK may only be shared for firmware updates for those IoT SE products, which can install/execute identical SE FW Update Images; for IoT SE products which cannot, different product-specific public SE-FAKs shall be used by the IoT SE developer.

If SE-FAK is a secret key (for verification of a MAC), the IoT SE developer shall choose it device-individually, either by randomly generating SE-FAK per device or by deriving SE-FAK per device, and keep SE-FAK and its related key derivation key (if any) confidentiality-protected in their development environment. A key derivation key may only be shared for deriving SE-FAK for firmware updates for those IoT SE products, which can install/execute identical SE FW Update Images; for IoT SE products which cannot, different product-specific key derivation keys for derivation of SE-FAKs shall be used by the IoT SE developer.

7.2.3 Package "Secure Update" – Addendum to security objectives rationale

Security objectives / Threats, OSPs and Assumptions from SPD	O.SE.SecureUpdate	OE.SE.FirmwareKeys
OSP.SE.SecureUpdate	X	
OSP.SE.StrongCrypto	X	
A.SE.FirmwareKeys		X

Table 9: Coverage of additional SPD items by the security objectives

OSP.SE.SecureUpdate is directly enforced by **O.SE.SecureUpdate** (which is a re-statement of **OSP.SE.SecureUpdate** as stated in this package)**.**

Concerning cryptographic functions used for authenticity-protection and encryption of firmware update images, **OSP.SE.StrongCrypto** is directly enforced by **O.SE.SecureUpdate**, which states that the corresponding authenticity-protection mechanism and the encryption shall have a security level of at least 100 bit.

A.SE.FirmwareKeys is directly upheld by **OE.SE.FirmwareKeys** (which is a re-statement of **A.SE.FirmwareKeys** as stated in this package).

7.3 Package "Secure Update" – Additional security requirements

7.3.1 Package "Secure Update" – Additional security functional requirements

FDP_ACC.1/SEFW Subset access control

Dependencies: FDP_ACF.1 Security based access control

FDP_ACC.1.1/SEFW **The TSF shall enforce the *IoT SE firmware update policy*[xxii] on**
(1) objects: SE FW update image, SE-FAK, SE-FCK;
***(2) operations: SE FW update, SE FW key update*[xxiii].**

FDP_ACF.1/SEFW Security attribute based access control

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/SEFW **The TSF shall enforce the *SE firmware update policy*[xxiv] to objects based on the following:**
(1) objects: SE FW update image, SE-FAK, SE-FCK;
***(2) attributes: SE-FAK signature/MAC, SE FW update version, and Latest SE FW version*[xxv].**

FDP_ACF.1.2/SEFW **The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:**
(1) SE FW update is allowed, if the SE-FAK signature/MAC is successfully verified against the corresponding SE FW update image and SE FW update version presented in the SE FW update request;
(2) SE FW key update is allowed, if the SE-FAK signature/MAC is successfully verified against the corresponding new SE-FAK and/or the new SE-FCK presented in the SE FW key update request[xxvi]**.**

FDP_ACF.1.3/SEFW **The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none***[xxvii]**.**

FDP_ACF.1.4/SEFW **The TSF shall explicitly deny access of subjects to objects based on the following additional rules: SE FW update is denied, if the SE FW update version presented in the SE FW firmware update request is older than the Latest SE FW version**[xxviii]**.**

AN(FDP_ACF.1/SEFW) The dependency to FMT_MSA.3 is not applicable. There are no default values for the attributes of this access control policy.

Remark: By enforcement of the explicit deny rule it shall be prevented that an attacker, by just applying a signed/MAC-protected SE firmware update image as officially released by the SE developer, can downgrade the SE firmware to an older version (e.g., to undo security fixes that were introduced in a newer SE firmware version). Still, the SE developer would have the ability to revert the SE firmware back to an older release (e.g., in case a newly issued firmware release shows problems or errors), by creating a new signature/MAC over the SE firmware update image of the older release together with some newer version number (which would be just introduced to enable this intended firmware downgrading).
Downgrading protection concerning SE-FAK and SE-FCK is not necessary, as an old key update request cannot be replayed successfully once the SE-FAK has been updated in the TOE.

FCS_COP.1/SE-FAK Cryptographic operation

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SE-FAK **The TSF shall perform *signature/MAC verification using SE-FAK***[xxix] **in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

AN(FCS_COP.1/SE-FAK): With all operations performed the resulting cryptographic operation has to provide a security level of at least 100 bit.

The dependencies to [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4 have to be resolved in the Security Target as the PP does not intend to additionally restrict the variety of product implementations and use cases.

FCS_COP.1/SE-FCK Cryptographic operation

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SE-FCK **The TSF shall perform *decryption using SE-FCK*[xxx] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

AN(FCS_COP.1/SE-FCK): With all operations performed the resulting cryptographic operation has to provide a security level of at least 100 bit.

The dependencies to [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4 have to be resolved in the Security Target as the PP does not intend to additionally restrict the variety of product implementations and use cases.

7.3.2 Package "Secure Update" – Additional security assurance requirements

ALC_FLR.1 (Basic flaw remediation) shall be augmented to the chosen SARs if this package is used.

7.3.3 Package "Secure Update" – Addendum to security requirements rationale

The following table shows that all additional SFRs chosen trace back to the additional TOE security objective:

TOE security objective / Security functional requirements	O.SE.SecureUpdate
FDP_ACC.1/SEFW	X
FDP_ACF.1/SEFW	X
FCS_COP.1/SE-FAK	X
FCS_COP.1/SE-FCK	X

Table 10: Tracing back SFR to TOE security objective for Package "Secure Update"

The following table shows that the package SFRs meet the package TOE security objective:

TOE security objective	SFR	Rationale
O.SE.SecureUpdate	FDP_ACC.1/ SEFW	Defines the requirement for a firmware update policy and defines the corresponding objects, which can be updated, and the update operations
	FDP_ACF.1/ SEFW	Defines the requirement for security attribute based access control for the update operations, the corresponding security attributes and the rules allowing only authentic images and keys to be updated, and preventing downgrading
	FCS_COP.1/	Defines the requirement for a cryptographic operation

TOE security objective	SFR	Rationale
	SE-FAK	signature/MAC verification ensuring that only authentic SE FW update images or authentic SE FW key updates are accepted by the TOE
	FCS_COP.1/ SE-FCK	Defines the requirement for a cryptographic operation decryption ensuring that confidentiality of SE FW update images and in particular of SE FW key updates can be ensured

Table 11: Mapping of SFRs to TOE security objective for Package "Secure Update"

The following table shows that the dependencies arising from the additional SFRs are either satisfied within this PP or corresponding rationale is referenced (typically provided in SFR application notes AN), why a dependency is either not applicable at all or why it has been left open to be satisfied by the ST/PP developer:

SFR	Dependency	Satisfied in this PP?
FDP_ACC.1/SEFW	FDP_ACF.1	Yes (by FDP_ACF.1/SEFW)
FDP.ACF.1/SEFW	FDP_ACC.1 FMT_MSA.3	Yes (by FDP_ACC.1/SEFW) No, not applicable as qualified by AN(FDP_ACF.1/SEFW)
FCS_COP.1/SE-FAK	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No, qualified by AN(FCS_COP.1/SE-FAK) No, qualified by AN(FCS_COP.1/SE-FAK)
FCS_COP.1/SE-FCK	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No, qualified by AN(FCS_COP.1/SE-FCK) No, qualified by AN(FCS_COP.1/SE-FCK)

Table 12: Satisfaction of SFR dependencies for Package "Secure Update"

8 Annex

8.1 References

[IoT-SCM-PP] *IoT Secure Communications Module Protection Profile (IoT-SCM-PP)*, version 1.0.0, 2019-12-19, by Secure Communications Alliance (SCA).

8.2 Glossary

AAK	Admin Authenticity Key
ACK	Admin Confidentiality Key
AIS	Applications and Interpretations of the Scheme (by German BSI)
AN	Application Note
Authenticity	Provable property of data that data have been created by a specific originator and that the data have not been corrupted after its creation (the latter meaning that authenticity also covers integrity of the data)
CC	Common Criteria
EAL	Evaluation Assurance Level
IDK	IoT Device Key (i.e. a key stored/used in an IoT SE)
IoT	Internet of Things
LAN	Local Area Network
PP	Protection Profile
SAK	SE Authenticity Key
SCK	SE Confidentiality Key
SAR	Security Assurance Requirement
SCA	Secure Communications Alliance
SCM	Secure Communications Module
SE-FAK	SE Firmware Authenticity Key (optional for the IoT SE)
SE-FCK	SE Firmware Confidentiality Key (optional for the IoT SE)
SE	Secure Element
security level	The security level of a cryptographic mechanism is usually given as the number of operations necessary for an adversary to successfully break the security provided by the mechanism. It is expressed as a base 2 logarithm, e.g., 100 bits of security means that 2^{100} operations are necessary.[8]
SFR	Security Functional Requirement

[8] The reader may consult NIST SP 800-57 part 1, Tables 2 and 3, for a first orientation on the security level of some well-known cryptographic algorithms. The final rating of the security level as well as the principle suitability of certain cryptographic algorithms is up to the TOE's CC certification scheme, though.

8.3 Original SFR Operations as Defined in CC part 2

End notes (indicated by Roman numerals) on assignment and selection operations in SFRs in section 6.1, which have partially or completely been executed in this PP, will lead to the following original assignment or selection operation statements from CC part 2:

---

[i] **[assignment: *access control SFP*]**

[ii] **[assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]**

[iii] **[assignment: *access control SFP*]**

[iv] **[assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]**

[v] **[assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]**

[vi] **[assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]**

[vii] **[assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]**

[viii] **[assignment: *the authorised identified roles*]**

[ix] **[assignment: *list of TSF-mediated actions*]**

[x] **[assignment: *list of TSF mediated actions*]**

[xi] **[assignment: *list of management functions to be provided by the TSF*]**

[xii] **[assignment: *tampering scenarios*]**

[xiii] **[assignment: *list of devices*]**

[xiv] **[assignment: *types of emissions*]** (SFR not from CC, but from section 5.1 in this PP)

[xv] **[assignment: *list of types of TSF data*]** (SFR not from CC, but from section 5.1 in this PP)

[xvi] **[assignment: *list of types of user data*]** (SFR not from CC, but from section 5.1 in this PP)

[xvii] **[assignment: *type of users*]** (SFR not from CC, but from section 5.1 in this PP)

[xviii] **[assignment: *types of interfaces/ports*]** (SFR not from CC, but from section 5.1 in this PP)

[xix] **[assignment: *list of types of TSF data*]** (SFR not from CC, but from section 5.1 in this PP)

[xx] **[assignment: *list of types of user data*]** (SFR not from CC, but from section 5.1 in this PP)

[xxi] **[selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]** (SFR not from CC, but from section 5.1 in this PP)

[xxii] **[assignment: *access control SFP*]**

[xxiii] **[assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]**

[xxiv] **[assignment: *access control SFP*]**

[xxv] **[assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]**

[xxvi] **[assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]**

[xxvii] **[assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]**

[xxviii] **[assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]**

[xxix] **[assignment: *list of cryptographic operations*]**

[xxx] **[assignment: *list of cryptographic operations*]**