Fraunhofer
SIT

FRAUNHOFER INSTITUTE FOR SECURE INFORMATION TECHNOLOGY

Security Module for the Electric Vehicle Charging System

Proposal for a Protection Profile

ATHENE
National Research Center
for Applied Cybersecurity

Fraunhofer
SIT

Security Module for the Electric Vehicle Charging System

Proposal for a Protection Profile

Version 1.0, November 2019

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages

DELTA

Datensicherheit und -integrität in
der Elektromobilität beim Laden
und eichrechtkonformen Abrechnen

Security Module for the Electric Vehicle Charging System

Proposal for a Protection Profile

Version 1.0, November 2019

Authors:

Andreas Fuchs
Christoph Krauß
Norman Lahr
Richard Petri

Contact:

Prof. Dr. Christoph Krauß
Fraunhofer Institute for Secure Information Technology
Rheinstraße 75
64295 Darmstadt
Germany

Phone +49 6151 869-116
Fax +49 6151 869-224
E-Mail christoph.krauss@sit.fraunhofer.de

This work has been partially funded by the German Federal Ministry of Economics and Technology (BMWi) within the project ”Datensicherheit und -integrität in der Elektromobilität beim Laden und eichrechtskonformen Abrechnen” (DELTA).

Imprint

Contact:
Fraunhofer Institute for Secure Information Technology SIT
Rheinstrasse 75, 64295 Darmstadt, Germany
Phone +49 6151 869-100
E-Mail info@sit.fraunhofer.de
URL www.sit.fraunhofer.de

Bibliographic information published by Die Deutsche Nationalbibliothek.
Die Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the internet at www.dnb.de.
ISSN 2192-8169

Christoph Krauß (ed.)
SIT-TR-2019-03: Security Module for the Electric Vehicle Charging System
Andreas Fuchs, Christoph Krauß, Norman Lahr, Richard Petri

Contents

1 PP Introduction

1.1 Introduction

In the field of e-Mobility, Electric Vehicles (EVs) are charged using a Charge Point (CP). To enable a trustworthy and reliable charging and billing process for the transferred electrical energy, data is exchanged between the vehicle and the CP through the charge cable and specific protocols are applied. Further, the CP is part of an infrastructure that connects the point on the one hand to the energy grid and to an energy provider, billing system, and other value-added service providers on the other hand. Value-added service providers are services that are not directly required for a charging process but may offer additional benefits for a customer. Examples for such services are the reservation of a charge point, network operator service, fleet managing, vehicle manufacturer service, and services such as mapping services.

The international standardisation system has created a basis for communication between an electric vehicle and the charging infrastructure in the form of international standard ISO/IEC15118, which is already in place. ISO/IEC15118 defines the communication between the electric vehicle and the CP. However, some protocols that are required for the value-added services are extensions of ISO/IEC15118.

On the vehicle side the protocols defined in ISO/IEC15118 are implemented in the Electric Vehicle Communication Controller (EVCC). The counterpart on the charging point side is realized in the Supply Equipment Communication Controller (SECC). Each of the communication controllers, EVCC and SECC, described before are using a Security Module (SecMod) that is the TOE in this document.

The TOE is an electronic unit that provides security related services to the EVCC and SECC, and is embedded in the environment detailed in Figure 1.1 (the TOE is shown as SecMod). The services are, for instance, cryptographic primitives, secure key generation and storage, and random number generation, that supports the security functions in the communication protocols. Furthermore, the TOE provides features to secure the EVCC and SECC on the system level and to proof the trustworthiness to third-parties. Therefore, the TOE provides mechanisms to enable secure and authentic boot, remote attestation, secure update process, and secure I/O.

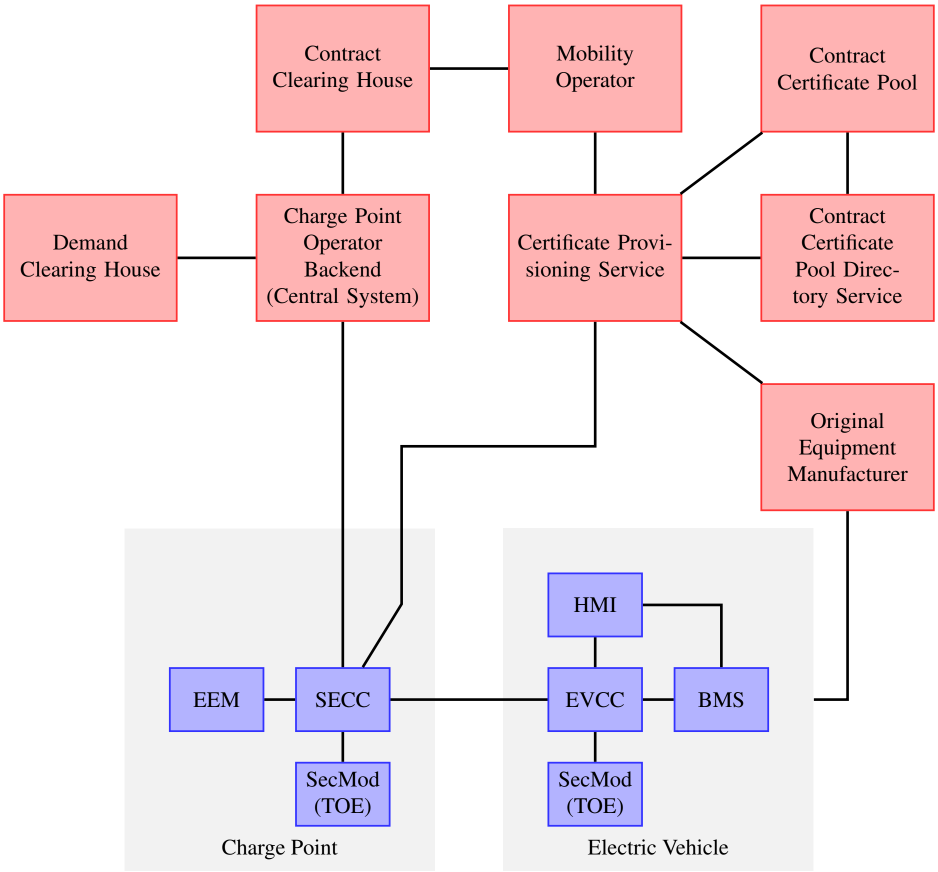
Contract Clearing House
Mobility Operator
Contract Certificate Pool
Demand Clearing House
Charge Point Operator Backend (Central System)
Certificate Provisioning Service
Contract Certificate Pool Directory Service
Original Equipment Manufacturer
HMI
EEM
SECC
EVCC
BMS
SecMod (TOE)
SecMod (TOE)
Charge Point
Electric Vehicle

Figure 1.1: System architecture including the TOE, its direct environment, and the infrastructure.

1.2 PP Reference

Title: Security Module for the Electric Vehicle Charging System

Version: 1.0

Date: Thursday 19th December, 2019

Authors: Andreas Fuchs, Christoph Krauß, Norman Lahr, Richard Petri

Keywords: e-Mobility, Electric Vehicle Communication Controller, Supply Equipment Communication Controller, Security Module, Protection Profile

1.3 Specific Terms

Term	Definition
CCH, Contract Clearing House	The Contract Clearing House is responsible to handle the collaboration between Charge Point Operators (CPOs) and Mobility Operators (MOs). The CCH is contacted by the central system of a CPO if an EV driver wants to charge her EV at a charge point, that does not belong to the MO he has an active contract with. In this so called roaming scenario the CCH has contracts with both the CPO and the MO and helps to direct the authorization requests and charging data, so called Charge Detail Record (CDR)s, to the respective MO. The CCH is also referred to as eRoaming Platform or E-Mobility Operator Clearing House. The resulting network of all participating CPOs and MOs is called an Intercharge Network. [4, 7]

Term	Definition
CCP, Contract Certificate Pool	Because ISO 15118 requires certificate installation and update to be finished within a very strict time limit (approx. 5 seconds, the Certificate Provisioning Service (CPS) is typically contacted by the MO in advance with the respective data to generate the response message for contract certificate installation and update. After the response messages were generated they are send to the Contract Certificate Pool (CCP). The CCP stores these response messages until they were requested once and deletes them afterwards. The response messages are stored with a reference to the respective Provisioning Certificate Identifier (PCID) or E-Mobility Account Identifier (EMAID), which identifies the vehicle the contract certificate belongs to, and potentially registered with a Contract Certificate Pool Directory Service (CCPDS). As stated in, there may be more than on CCP which makes the use of a directory service necessary to make other relevant and authorized stakeholders aware on where to find the respective response messages. The role can be assumed by any participant including Contract Clearing House (CCH), CPO, MO, Original Equipment Manufacturer (OEM), or a third party provider. The CCP may also provide additional services like validity verification of data or notification services for other stakeholders like the previously mentioned CCH, CPO, MO, or OEM. These additional services are not further defined in. [8]
CCPDS, Contract Certificate Pool Directory Service	This entity is responsible to operate and maintain the Contract Certificate Pool Directory Service. This service is consumed during contract certificate installation and update by the CPO to find the CCP that contains the prepared contract certificates for a given customer and contract number (EMAID). This role may be assumed by MOs, CPOs, CCHs, or any other third party entity. [8]

Term	Definition
CDR, Charge Detail Record	–
ContractCert, Contract Certificate	–
CP, Charge Point	A Charge Point is an electric vehicle charging station, also called EV charging station, electric recharging point, charging point, ECS (electronic charging station), and Electric Vehicle Supply Equipment (EVSE). It is an element in an infrastructure that supplies electric energy for the recharging of plug-in electric vehicles—including electric cars, neighborhood electric vehicles and plug-in hybrids.
CPID, Charge Point Identifier	–
CPO, Charge Point Operator	The Charge Point Operator is the entity that operates and manages charge points. The CPO may also be the manufacturer of the charge point, but this task could also be off-loaded to an external charge point manufacturer. The on-site maintenance of charge points may also be off-loaded to an external service provider as well. The CPO can also be split into Sub Operator and Hub Operator and is also referred to as EVSE Operator. The role is also sometimes divided into the business operator and technical operator role, which further emphasizes the off-loading aspect of manufacturing and maintenance. The CPO is also responsible to operate and maintain the so called CPO Sub-CA 1 and 2 Certificate Authoritys (CAs), which are necessary to generate the EVSE Leaf Certificates for the respective CP. [8, 7, 4, 5]

Term	Definition
CPS, Certificate Provisioning Service	The Certificate Provisioning Service is necessary during the installation and update of contract certificates within an EV. According to ISO 15118 it is a secondary actor which can be assumed by a CPO, MO, a third party provider, or the EVCC itself. The CPS is equipped with a Leaf Prov Certificate and its respective chain up to a Prov Sub-CA 1 certificate. In case a contract certificate is installed or updated, the provisioning service is used to sign the contract certificate chain as well as the encrypted private key, DH Public key and EMAID from the respective MO to enable the EV to verify the integrity and authenticity of aforementioned mentioned data. Before the data is signed, the service also needs to verify the validity of the data provided by the MO. Once the data is signed, the resulting response messages for certificate update and installation are send to the CCP. [5, 8]
ECU, Electronic Control Unit	An EV contains many Electronic Control Units (ECUs) with very different functionalities. An Electronic Control Unit (ECU) is a controller that may implement certain functionality with regards to the charging of an EV and also can provide important information about an EV. [4]
EMAID, E-Mobility Account Identifier	–

Term	Definition
EV, Electric Vehicle	The Electric Vehicle describes a typical vehicle that is owned by an EV owner. The current EV driver at some point wants to charge the EV at a charge point. Once the EV is connected to the charge point it will try to communicate and charge using the protocol specified in ISO 15118. The EV is equipped with its own unique OEM Provisioning Certificate, a Contract Certificate specifically issued to the EV owner, and several Vehicle to Grid (V2G) Root Certificates (and possibly MO Root CA). The associated OEM is responsible to install the OEM Prov Certificate and is also responsible to manage the EV in regards to certificate installation. [4, 5, 6]
EVCC, Electric Vehicle Communication Controller	The Electric Vehicle Communication Controller is part of the EV and implements the ISO 15118 communication and all necessary utilities. [4, 5]
EVCCID, Electric Vehicle Communication Controler Identifer	Specifies the EV's identification in a readable format. [4]
EVSE, Electric Vehicle Supply Equipment	The Electric Vehicle Supply Equipment is a different term to describe a charge point for Electric Vehicles. The EVSE is equipped with a EVSE Leaf Certificate and the respective certificate chain up to a CPO Sub CA-1 which is used to authenticate the EVSE to an EV. The EVSE belongs to a specific CPO and is managed by it. [4, 6]

Term	Definition
MO, Mobility Operator	The Mobility Operator (MO) is typically a service provider that has a contract relationship with an EV owner that allows the current EV driver to charge the EV at a charge point. To charge the EV at the charge point, the charge point must either belong to the MO or it must support the roaming scenario. The MO can be an electricity provider or a general service provider that is selling energy as an intermediary. The MO Sub-CA 1 and MO Sub-CA 2 are also operated and maintained by the MO. These are necessary to generate the contract certificate stored in the EV owner's car to enable ISO 15118 based communication and charging. The MO is also referred to as Emobility Service Provider which can be further separated into Sub Providers and Hub Providers, or as an E-Mobility Operator. [4, 5, 8, 7]
OEM, Original Equipment Manufacturer	The Original Equipment Manufacturer refers to the manufacturer of the EV. The OEM may also optionally fulfil the role of an MO. The OEM is also responsible to operate and maintain the OEM SUB-CA 1 and OEM Sub-CA 2 Certificate Authorities, which are needed to generate the OEM Provisioning Certificate for the EV. [4, 8]
OEMPC, OEM Provisioning Certificate	Certificate issued to the EVCC to enable the provisioning of a contract certificate. It is securely requested and received from a secondary actor to uniquely identify the EVCC. [4]

Term	Definition
RTM, Root of Trust for Measurement	The TOE supports the integrity measurement of the trusted platform by calculation and reporting of measurement digests of measured values. Typically the Root of Trust for Measurement (RTM) is controlled by the Core Root of Trust for Measurement (CRTM) as the starting point of the measurement. The measurement values are representations of embedded data or program code scanned and provided to the TOE by the measurement agent. The TOE supports cryptographic hashing of measured values and calculates the measurement digest by extending the value of a Platform Configuration Register (PCR) with a calculated or provided hash value. The PCRs are shielded locations of the TOE which can be reset by TOE reset or a trusted process, and written only through measurement digest extensions.
RTR, Root of Trust for Reporting	The TOE holds the Endorsement Primary Seed (EPS) and generates Endorsement Keys (EK) from the EPS. The EK and the corresponding Endorsement Certificates define the trusted platform identities for the Root of Trust for Reporting (RTR). The TOE may be shipped with an EK and a Certificate of the Authenticity of this EK. The EK is bound to the Platform via a Platform Certificate, providing assurance from the certification body of the physical binding and connection through a trusted path between the platform (the RTM) and the genuine TOE (the RTR). The attestation of the EK and the Platform Certificates build the base for attestation of other keys and measurements.

Term	Definition
RTS, Root of Trust for Storage	The TOE holds the Storage Primary Seed (SPS) and generates Storage Root Keys (SRK) from SPS. The SRK are roots of Protected Storage Hierarchies associated with a TOE. One use of the storage keys in these hierarchies are used for symmetric encryption and signing of other keys and data together with their security attributes. The resulting encrypted file, which contains header information in addition to the data or the key, is called a BLOB (Binary Large Object) and is output by the TOE and can be loaded in the TOE when needed. The private keys generated on the TOE can be stored outside the TOE (encrypted) in a way that allows the TOE to use them later without ever exposing such keys in the clear outside the TOE. The TOE uses symmetric cryptographic algorithms to encrypt data and keys and may implement asymmetric cryptographic algorithms of equivalent strength.
SECC, Supply Equipment Communication Controller	The Supply Equipment Communication Controller belongs to the EVSE and implements the ISO 15118 communication and all necessary utilities. It is also sometimes referred to as Local Controller. [4, 5, 6]
SECCCert, Supply Equipment Communication Controller Certificate	–
SecMod, Security Module	A Security Module is utilized by an EVCC and SECC for cryptographic support, e. g., realized in form of an Hardware Security Module. The requirements for the Security Module are defined in this Protection Profile.
SRK, Storage Root Key	A cryptographic key, usually used by Hardware Security Modules (HSMs) to encrypt other key material to store it outside of the secure storage of the HSM

Term	Definition
VAS, Value Added Service	Value Added Services describe services that are not directly required for a charging process but may offer additional benefits for a customer. An example for such a VAS is the reservation of a CP. [4]

1.4 TOE Overview

1.4.1 Introduction

The TOE as defined in this Protection Profile is the Security Module (SecMod) which is an electronic unit closely connected to or contained in the EVCC or SECC. The EVCC is the control unit in an EV and responsible for the communication with the SECC in a Charge Point to manage the charging and billing processes in the charging system infrastructure (cf. Figure 1.1). The purpose of the TOE is to provide security related services to either EVCC or SECC. In the following, the overall charging system infrastructure is described in Section 1.4.2 and the functionality and interfaces of TOE itself are detailed in Section 1.4.3.

1.4.2 Description of the Charging Infrastructure

In this section the application environment of the TOE is described. At the top, the intended application environment is a charging infrastructure for Electric Vehicles, including Battery Electric Vehicles and Plug-In Hybrid Vehicles, that feature the so called Plug-and-Charge (PnC) functionality as the main use case. Figure 1.1 shows a general system architecture. PnC is introduced by ISO/IEC15118 [4]. ISO/IEC15118 specifies the communication protocols between an EV and a CP, also referred to as EVSE, to enable a fully automated billing process for charging an EV over a CP. This process is transparent to a customer. She has just to connect her vehicle to a CP with a plug. This includes the load control, identification, and authorization of the customer so that no further intervention is required. In addition, the infrastructure provides Value Added Services (VASs) that enable extra functionality which are out of scope of ISO/IEC15118. For instance, VASs can be a remote update for the EV or Internet access during the charging process.

As shown by Figure 1.1, the actual ISO/IEC15118-based communication between an EV and a CP is handled by the EVCC on the EV side and by the SECC on the CP side. The EVCC and the SECC are ECUs. The TOE is closely coupled (or even contained) in the EVCC and SECC, and offers cryptographic services in a manner similar to an HSM. The automatic charging and billing process is based on digital certificates [4]. This includes the specification of a Public Key Infrastructure (PKI) and a certificate hierarchy in [5].

Within this PKI, each EVCC is supplied with a OEM Provisioning Certificate that includes a unique PCID, which uniquely identifies an EV. The validity period of a OEM Provisioning Certificate is aimed to match the lifetime of the EV and is, under normal circumstances, never replaced. Further, the EVCC is uniquely identified with the Electric Vehicle Communication Controler Identifer (EVCCID) in the low level communication with the SECC. The initial actor is the *customer* who is a person having access to an EV. Typically, she contracts a MO as an energy supplier for her EV. This contract is linked to a digital Contract Certificate that uniquely identifies the customer by the contained unique EMAID. The validity period of Contract Certificates is short (1-24 months) and they are renewed on a regular basis.

The CPO is the legal entity that deploys, manages, and maintains CPs and is typically an energy supplier. The CPO is also responsible to operate and maintain the CPO Sub-Certificate Authorities (1 and 2), which are necessary to issue the Supply Equipment Communication Controller Certificate for the respective CPs. A Supply Equipment Communication Controller Certificate contains a unique Charge Point Identifier (CPID) including location information to identify a CP. The charging infrastructure provides so called roaming if customers charge their EVs at CPs that do not belong to their contracted MO. The roaming is handled by the CCH. This entity supports a CPO by mapping a customer to her MO by utilising her Contract Certificate.

1.4.3 The TOE in the EVCC and SECC

The SecMod or TOE, respectively, is an independent subsystem that supports the EVCC and SECC with cryptographic services. It consists of a control unit that controls the data processing, memory for secure storage, cryptographic primitives and protocols implemented in software and, optionally, accelerated by hardware co-processors, and enhanced security features.

The cryptographic services are accessible via a physical or logical interface (cf. 1.4.4). The access to the TOE is restricted via an access control mechanism. The channel between the TOE and another accessing component is established securely so that the security requirements in 5 can be fulfilled. According but not limited to the network and application protocol requirements listed in ISO/IEC15118-2 [5] the Security Module must feature the following:

- Digital Signature Schemes,
- Cryptographic Hash Functions,
- Secure Key Generation,
- True Random Number Generation,

- Secure Storage for Keys and Data,
- Secure Data Storage,
- Secure Key Migration,
- Monotonic Counter(s),
- Only-extend Registers (Platform Configuration Register),
- Remote Attestation Protocol,
- Direct Anonymous Attestation Protocol (only applicable to EVCC),
- Key Policy Enforcement and Access Control,
- Secure Host Communication with mutual authentication.

True Random Number Generation, as well as Secure Key Generation are general required features for any Hardware Security Module. Both are used, for example, to generate key material for signature schemes. The Digital Signature Schemes are required to support remote attestation protocols. Furthermore, the EVCC utilizes digital signatures to implement functionalities concerning the Contract Certificate, e. g., the TOE is supports the secure use of the corresponding private keys. In case of the SECC, such signatures are used to implement the TLS protocol, e. g., the TOE enables the secure use of the certificates and corresponding private keys. Cryptographic hash functions are a required feature, e. g., to support HMAC protocols, session integrity, or access control. The SecMod should support an agile communication interface to enable the quick substitution of cryptographic primitives and their parameters.

Apart from the provided cryptographic services, the SecMod serves as a root of trust for storage and reporting. As such, the SecMod must be able to securely store data, especially key material. This data is not necessarily stored on memory inside of the SecMod, it may also utilize a Storage Root Key (SRK) (or alternatively a seed from which such a key can be generated). The SRK is then used to securely store keying material outside of the SecMod in an encrypted form, only decryptable by the SecMod's SRK. The SRK enables the safe storage of the key material created in the context of ISO/IEC15118. This include, at minimum, the private parts of the key pairs associated with the Contract Certificates and OEM Provisioning Certificate (OEMPC), and optionally the certificates themselves up to the root. The only-extendable registers (or PCR) can support a root of trust for measurement by storing platform measurements. SecMod serves as root of trust for reporting and can thus used to carry out remote attestation. SecMod supports Direct Anonymous Attestation (DAA) to preserve privacy of the platform's user.

Lastly, the SecMod supports an extensive policy and access control scheme to its functionalities. It offers secure host communication with mutual authentication, for example, by password protection or asymmetric encryption. Furthermore, access to keys can be controlled with a policy scheme. Keys may be protected by PINs or other authentication values, bound to measured information such as PCRs and values of monotonic counters, authorized by external signature, or otherwise a logical combination of these conditions.

1.4.4 TOE Type

The SecMod provides extended security services for cryptographic operations and related services (such as counters, key management and storage, or policy enforcement) as described in Section 1.4.4. These services are used by the EVCC and SECC akin to an HSM. Multiple types of implementations of a SecMod are possible:

Discrete HSM The SecMod may be a discrete HSM, i. e., a separate chip dedicated to the security functionality with an external communication interface used by the EVCC and SECC to communicate with the SecMod.

Integrated Implementation The SecMod may be implemented with specialized hardware integrated into the application processor used by the EVCC and SECC, including an internal communication interface.

Firmware Implementation The SecMod may be implemented by software running in a Trusted Execution Environment (TEE) offered by the application processor used by the EVCC and SECC, including a secure Application Programming Interface (API), which offers a controlled communication across the boundary of the TEE.

While these three variants all offer a different interface to the application, the implementation should provide a Hardware Abstraction Layer (HAL) to enable the transparent communication with the SecMod. This standardized HAL can then be used by the supplementary software of the SecMod in a manner independent of the actual implementation type.

1.4.5 TOE Physical Boundary

The SecMod comprises the hardware and software relevant for the security functionality used by the EVCC and SECC as defined in this PP.

Note, that the SecMod is physically embedded into the EVCC, which itself is an ECU of an EV. Accordingly, the same level of physical protection can be assumed as is for the environment of the EV. In the case of the SECC, the SecMod is physically embedded into the electronic components of a CP. Hence, the same level of physical protection can be assumed as for the environment of the CP.

1.4.6 TOE Logical Boundary

The logical boundary of the TOE is defined by the HAL (cf. Section 1.4.4) providing the communication interface. The TOE must be able to handle any valid or malformed type of communication via this interface, without exhibiting undefined behaviour.

1.5 TOE Life Cycle Model

The life cycle of the TOE is similar to that of other HSMs, comprising 5 phases, as well as potential sub-phases and variants depending on the implementation type.

Development The initial phase is the development of the SecMod. Depending on the implementation type (cf. Section 1.4.4), this may consist of several, possibly overlapping, sub-phases. In case of a hardware-based implementation, the Integrated Circuit (IC) must be developed or chosen from an existing HSM, to support the required feature set. Furthermore, in case of an integrated implementation, the IC compromising the TOE must be integrated into an existing application processor. In case of a firmware-based implementation, an application processor with an appropriate TEE must be chosen. In all cases, the SecMod firmware must be developed.

Manufacturing After the SecMod soft- and hardware is developed, the manufacturing may begin. In case of hardware-based implementations, the IC is manufactured, personalized, loaded with (possibly an initial) firmware and tested. Firmware-based implementations may require a specialized deployment procedure for the firmware, depending on the chosen application processor. For example, the setup of a PKI or other key hierarchy might be necessary for digitally signing the firmware. The manufacturing may also include the generation of initial seeds and keys (e. g., internal storage keys).

Integration The manufactured SecMods can then be delivered, deployed and integrated into the EVCC or SECC. Apart from the installation of the hardware in the case of hardware-based implementations (e. g., soldering the SecMod dedicated chip), this includes the generation of initial key pairs and certificates (and/or certificate requests) in the context of the ISO/IEC15118 key hierarchy.

Operation The operational phase begins with the finalized integration of the SecMod and the operation of the EVCC or SECC.

Decommission As the components in the context of ISO/IEC15118 all have a limited lifespan, the life cycle should also include a final decommission. This mostly involves the destruction of key material to avoid abuse.

The scope of the evaluation of the TOE only covers the development and manufacturing phase (or the results thereof) up until the delivery of the TOE. This includes all hard- and software concerning the TOE. In the case of a firmware-based implementation, an evaluation may cover a specific firm- and hardware combination.

2 Conformance Claims

This document is a draft for a Protection Profile for a Security Module for the Electric Vehicle Charging System. It serves as a baseline for a Protection Profile document conformant with the Common Criteria version 31̇ Release 5.

3 Security Problem Definition

The following sections describe the security problem definition of this Protection Profile.

3.1 Assets

This section lists the assets that are related to the TOE and that has to be protected against misuse. Furthermore, assets of the EVCC/SECC that have to be protected by the TOE are listed. The required protection can be confidentiality (C), integrity (I), authenticity (A), and privacy (P).

3.1.1 System-related Assets

This section lists the assets that has to be protected by the TOE.

CCH Authenticiation Data History (C,I,P) Next to the currently valid Authentication Data, the CCH may also store a history of the provided MO - CCH Authentication Data provided by the MO. This asset is originally described in: [7].

CCH EVSE Status Record (C,I,A) The CCH allows CPOs to upload status information about their charge points. MOs can request this data from the CCH dynamically. The CCH will thus also store this information in its database. The data will be dropped if an EVSE is out of order, free, or occupied, for example. This asset is also referred to as: Point of Interest (POI) Data [7]. This asset is originally described in: [7].

Certificate Update Retry Counter (C,I,A,P) Encodes after how many days the EVCC should try to send another CertificateUpdateReq message, if the previous message failed due to various reasons. It can thus be assumed that the value is stored on the EV in some capacity. Sent as part of the CertificateUpdateRes message. This asset is also referred to as: RetryCounter [5]. This asset is originally described in: [4, 5].

Challenge Challenge generated by the CP, or more specifically the SECC, and sent to the EVCC as part of the PaymentDetailsRes during the payment negotiation. This challenge is then signed by the EVCC and sent back in the follow up message called AuthorizationReq to make sure the Contract Certificate belongs to the current EV connected to the CP. Has to be stored at least until the follow up message has been received. ISO15118-2 does not define this number as a nonce! This asset is also referred to as: GenChallenge [5]. This asset is originally described in: [4, 5].

ChargeBoxSerialNumber (I,A) Unique Identfier for the charge box inside a CP. This asset is originally described in: [6].

Charge Detail Record (C,I,A,P) Contains the aggregated data of a charging session. It is generated by the CP or the respective CS of the corresponding CPO. It is send to the corresponding MO via the use of a CCH. The MO uses this aggregated data to bill the respective customer. The CDR format and informationcontained within is not yet standardized (neither legally nor technically) and depends on the used protocol. Currently it is defined by the CCH. In case of OICP It typically contains: Signed (optional!) meter values (start, end, and / or intermediate values), the amount of consumed energy, EVSE IDs, identification / authentication data (such as the EMAID, RFID UIDs, and QR Codes and may additonally also contain PINs), charging session data (start and end time), and further CCH related data. These may include session data such as session IDs (of the CCH connection), foreign session IDs (possible session IDs between any pair of CPO, EVSE, and MO or even internal identifiers) as well as Identifiers for CPO and MO within the CCH database. This asset is also referred to as: Service Detail Request (SDR) [4]. This asset is originally described in: [7], [4].

ChargePointModel (I,A) Depicts information on the model of a CP. This data is stored on the CP and can be accessed by the CPO. May be used to identify a CP. This asset is originally described in: [6].

Charge Service (C,I,A,P) Denotes the supported charge service that a CP supports. Transfered during the ServiceDiscoveryRes. Contains the SupportedEnergyTransferMode in addition to all other fields that a typical service (see Service List) contains. Is also stored on the CP. This asset is also referred to as: ChargeService [5]. This asset is originally described in: [4, 5].

Charging Profile (C,I,A,P) Contains serveral Profiles each consisting of ChargingProfileEntryStart, ChargingProfileEntryMaxPower, and ChargingProfileEntryMaxNumberOfPhasesInUse. This asset is also referred to as: ChargingProfile [5]. This asset is originally described in: [4, 5].

Charging Session (C,I,A,P) Encodes whether the EVCC wants the SECC to stop or pause the current charging session. Sent as part of the SessionStopReq. Must be stored as part of the session context on the EVCC and SECC. This asset is also referred to as: ChargingSession [5]. This asset is originally described in: [4, 5].

Contract Certificate (C,I,A,P) The Contract Certificate is issued by MO Sub 2 CAs and is used by the EV to request authorization to start a charging session. The certificate is generally stored inside the EVCC and acts as a unique identifer for the EV / customer. The certificate contains information on Issuer (Country, Organization, Common Name) and Subject (Organization, Common Name (EMAID), and Domain Component) among other things. According to AR E 2802-100-1, when the EV requests a certificate update, a new contract certificate is generated by the MO and stored inside a CCP together with a respective link to its location inside a CCPDS. The Contract Certificate is stored there by the CPS after it received the prepared CertificateInstallation and CertificateUpdate request messages. Inside these messages it is stored together with the respective private key (which is stored inside an encrypted PKCS #12 container) in the CCP until either CPO or OEM (depending on the chosen installation path) requests it for its final delivery. This asset is originally described in: [4, 5].

Contract Certificate Private Key (C,I,A,P) Private key corresponding to the Contract Certificate. Required to request a certificate update request. It is stored inside the EVCC (or at least EV) and uniquely identifes the customer and EV. According to AR E 2802-100-1 it is also stored (though encrypted) or passed through the CPS, MO Sub 2 CA / OEM backend, and CCP until it is passed through the CPO backend to the CP to the vehicle. This asset is originally described in: [4, 5].

Contract Data (I,A) When VDE-AR-E 2802-100-1 talks about contract data, it talks about the combination of Contract Certificate and associated certificate chain, the respective Private Key. Signed by CPS and given to CCP. This asset is originally described in: [8].

Contract Signature Certificate Chain (C,I,A,P) Certificate chain of a contract certificate. Contains the contract certificate and all intermediate certificates, but not the root certificate. Sent as part of the CertificateInstallationRes, CertificateUpdateRes, CertificateUpdateRes, and PaymentDetailsReq. This asset is also referred to as: ContractSignatureCertChain [5]. This asset is originally described in: [4, 5].

CP Authorization Cache (C,I,A,P) Depending on the implementation, a charge point might store the currently active and previously authorized identifiers together with an expiriy date and its state of validity. This cache is managed by the CP itself. This asset is originally described in: [6].

CP Charging Profile (I,A) The CPO is able to configure its CP with so called charging profiles. These define how much energy the CP can leverage in a specific time interval. In case of OCPP, the Charging Profile consists of a schedule (which defines specific periods and charging parameters such as periods, minimum required charging rate and power limit) as well as a purpose, type, validity period, and recurrency. The Charging Profiles are identified using a unique identifier (ChargingProfileID) which is unique within a CPOs ecosystem. They may also be provided with a priority in case several profiles are defined. The CP may also combine different Charging Profiles to generate a composite profile. This asset is originally described in: [6].

CP ConnectorID (I,A) Depending on the implementation and used protocol (like OCPP), each single connector of a charge point may be supplied with its unique identifier. This asset is originally described in: [6].

CP Connector Status (I,A) Each CP has a number of connectors that a customer can use to plug his EV into. The CPO may remotly lock and unlock this connector. The CP will store this data as part of its configuration to some extend This asset is originally described in: [6].

CP-CPO Messages (C,I,A,P) Messages exchanged during communication between CP and CPO. This asset is originally described in: [6].

CP Diagnostic Data (C,I,A,P) The CP will generate and probably store diagnostic data about its current state. This data can include typical logging data but is not limited to it. This data may be accessed by the CPO remotely or locally. This asset is originally described in: [6].

CP Firmware (C,I,A) Firmware of the CP. Has version information and status information attached to it as well. This asset is originally described in: [6].

CP Local Authorization List (C,I,A,P) Depending on the implementation, a charge point might store the currently active and previously authorized identifiers together with an expiriy date and its state of validity. This list is centrally managed by the CPO Backend System and Synchronized between all charge points. This asset is originally described in: [6].

CP Message Queue (C,I,A,P) In case a CP loses its connection to the backend it needs to cache all messages that need to be send to the CPO backend system (Central System). It may be necessary to make sure that messages follow a specific order and thus these messages are stored in a queue. This asset is originally described in: [6].

CPO CP Data (I,A) The CPO will store information about his currently deployed and maintained CPs. This information will range from typical CP information up to current charging session related information (e.g. Occupation / availability status, Errors, etc.).

CPO Timestamps (C,I,A,P) Some Messages during communication between CP and CPO will contain timestamps which relate to start of transactions or specific events This asset is originally described in: [6].

CP Reservations (C,I,A,P) The CPO can prepare CPs for reservations by a customer. The CP may even store a list of reservations for specific time slots. The lists may contain expiry dates, ReservationIDs, and Authentication Data (such as EMAIDs, RFID-Tags, etc.) and ParentIDs. This asset is originally described in: [6].

CP Time (I,A) Current time on the CP. This is a cruical configuration item on the CP as it is relevant for billing a customer accordingly. CP and CPO backend have to synchronize at times. Typically set after BootNotification.conf message. This asset is originally described in: [6].

CP Timestamp (C,I,A,P) Some Messages during communication between CP and CPO will contain timestamps which relate to start of transactions or specific events This asset is originally described in: [6].

CP Vendor Error Code (I,A,P) The manufacturer / vendor of CP may define his own set of error codes for diagnostic purposes and maintenance. These may be stored on the CP and the CPOs backend system and will be part of status communication messages. This asset is originally described in: [6].

CP Vendor Specific MessageID (I,A,P) In order to extend the possible communication API, protocols may allow vendors to define additional messages for CPs that vendors may use to implement specific requirements or features. OCPP for example enables vendors to define a MessageID that the backend system may interpret during a DataTransfer.req from CP to CPO backend system. The protocol also defines an error message in case the backend does not understand the specific MessageID. This could be used to identify CPs. The MessageID actually acts as an identifier for the type of communication. This asset is originally described in: [6].

Customer/EV Authorization Status (C,I,A,P) The CP may store information about the current authorization of the customer/EV to reduce its dependcy on the backend system for each transaction. This data may include uniquely identifying data such as EMAID, RFID tags, etc. as well as validity information and expiry dates. This asset is originally described in: [6].

Electric Vehicle Communication Controller Identifier ((C),I,A,P) The Electric Vehicle Communication Controller Identifier (EVCCID) is basically the MAC of the EVCC within the vehicle and is used during the low level connection establishment during ISO 15118 communication as part of the SessionSetupReq. It uniquely identifies the EVCC of a vehicle and can thus also be used to identify the EV. This asset is originally described in: [4, 5].

EMAID (C,I,A,P) The E-Mobility Account Identifier (EMAID) uniquely identifies the contract between a customer and the MO. The format consists of a two digit prefix that encodes the country, followed by three digits encoding the MO name, and finally 5 digits of MO internal account identifier. The EMAID is used by the MO to authorize charging sessions and handle the billing. The CCH uses the EMAID, more specifically its prefix, to determine the MO it needs to contact for an authorization request. This asset is also referred to as: Electric Vehicle Contract Identifier (EVCOID) [7]. This asset is originally described in: [4, 5], [7].

Encrypted Contract Certificate Private Key (C,I,A,P) Contains the encrypted private key for a newly generated contract certificate. It is encrypted using a ECDH protocol where the public key DHpublickey is also part of the message. Sent as part of the CertificateUpdateRes and CertificateInstallationRes messages. ISO15118-2 and VDE AR 2802-100-1 propose to generate it in the backend by the MO or OEM in advance to the CertificateUpdateReq and CertificateInstallationReq. While ISO15118-2 assumes the CCH will probably store the contract certificate and private key, VDE AR E 2802-100-1 assumes that the CCP stores them. It should be noted that both only propose this as a process for generation and installation but do not define it as the single valid solution. This asset is also referred to as: ContractSignatureEncryptedPrivateKey [5]. This asset is originally described in: [4, 5].

EV Charge Parameter (C,I,A,P) These are the summary of parameters related to the charging session that belong to the EV. These parameters are stored on the EV and communicated to the CP during the communication session. Depending on the implementation, the CP will probably store this information for the duration of the charging session and might even log the data. Subsets of these parameters are sent as part of the ChargeParameterDiscoveryReq, CurrentDemandReq, and PreChargeReq messages. The following list summarizes these parameters: AC_EVChargeParameter (Aggregated Datatype containing a subset of the other parameters), BulkChargingComplete, BulkSOC, ChargingComplete, DC_EVChargeParameter (Aggregated Datatype containing a subset of the other parameters), DC_EVStatus, DepartureTime, EAmount, EVEnergyCapacity, EVEnergyRequest, EVMaxCurrent / EVMaximumCurrentLimit, EVMaxVoltage / EVMaximumVoltageLimit, EVMinCurrent, EVTargetVoltage, EVTargetCurrent, FullSOC, MaxEntriesSAScheduleTuple, RemainingTimeToBulkSOC, RemainingTimeToFullSOC, RequestedEnergyTransferMode. This asset is also referred to as: AC_EVChargeParameter [5], DC_EVChargeParameter [5]. This asset is originally described in: [4, 5].

EV Charge Progress (C,I,A,P) Sent by the EVCC during the PowerDeliveryReq. Indicates the current charging state of the EV. This asset is also referred to as: ChargeProgress [5]. This asset is originally described in: [4, 5].

EV Charging Mode Status (C,I,A,P) Encodes the current status of the EV. While this data is exchanged during ISO 15118 communication, it should be noted that the EV stores this data (probably in another format) as well. Sent as part of the ChargingStatusReq for AC charging and WeldingDetectionReq, PreChargeReq, CableCheckReq and CurrentDemandReq for DC charging. This asset is also referred to as: DC_EVStatus [4, 5]. This asset is originally described in: [4, 5].

EV DC Power Delivery Paramters (C,I,A,P) Comprises the power delivery parameters for DC charging. Sent by the EVCC as part of the PowerDeliveryReq. This asset is also referred to as: DC_EVPowerDeliveryParamters [5]. This asset is originally described in: [4, 5].

EV Firmware (C,I,A,P) The firmware of the EV. It has a version information and status information attached to it as well.

EVSE Charge Parameter (C,I,A,P) These parameters are the summary of charging related parameters of the CP. These parameters are stored on the CP and communicated to the EV during ISO15118 communication as part of the ChargeParameterDiscoveryRes, ChargingStatusRes, CurrentDemandRes, PreChargeRes and WeldingDetectionRes messages. Depending on the implementation, the EV has to store the received parameters are part of the current session and might also log the data for further use. The following list summarizes the parameters: AC_EVSEStatus, DC_EVSEStatus, EVSECurrentLimitAchieved, EVSECurrentRegulationTolerance, EVSEEnergyToBeDelivered, EVSEMaxCurrent, EVSEMaximumCurrentLimit, EVSEMaximumPowerLimit, EVSEMaximumVoltageLimit, EVSEMinimumCurrentLimit, EVSEMinimumVoltageLimit, EVSENominalVoltage, EVSEPeakCurrentRipple, EVSEPowerLimitAchieved, EVSEPresentCurrent, EVSEPresentVoltage, EVSEVoltageLimitAchieved. This asset is also referred to as: AC_EVSEChargeParameter [5], DC_EVSEChargeParameter [5]. This asset is originally described in: [4, 5].

EVSEID (I) Uniquely identifies a CP. While there is no general specification for the format of such an identifier, it is possible that CPOs could encode location information into this identifer. This asset is also referred to as: ChargePointSerialNumber [6], Charge Point Identifier (CPID) [5]. This asset is originally described in: [5], [7].

EVSE Processing Status (C,I,A,P) Sent as part of the AuthorizationRes and CableCheckRes (for DC charging) message to indicate whether the CP has finished processing the request at the time of sending the message. It thus encodes status information. This asset is also referred to as: EVSEProcessing [5]. This asset is originally described in: [4, 5].

EVSE Timestamp (C,I,A,P) Sent as part of PaymentDetailsRes and SessionSetupRes . This asset is also referred to as: EVSETimeStamp [5]. This asset is originally described in: [4, 5].

Integrated Circuit Card Identifier (ICCID) (I,A) Many online CPs are equipped with a model that uses a Smart Card chip for authentication (at least in case of GSM for example). The ICCID is the unique identifier of the Smart Card used inside the modem for authentication. It can be used to identify the CP and may also be reported to the CPO backend for processing. This asset is originally described in: [6].

Identification Data (C,I,A,P) Unique identifier sent to the CCH by the CPO that uniquely identifies a customer. This unique identifier is used to authorize a charging session and can be either a RFID Mifare Token, EVCOID / EMAID, etc. In case of Plug and Charge, the EMAID is used as a means for the CCH to find the correct MO. This asset is also referred to as: IdToken [6]. This asset is originally described in: [7].

Internation Mobile Subscription Identitiy (IMSI) (I,A) Many online CPs are equipped with a model that uses a Smart Card chip for authentication (at least in case of GSM for example). The IMSI is the unique identifier used to identify participants inside a cellular network. Like the ICCID, it can be used to identify the CP and may also be reported to the CPO backend for processing. It is typically stored inside the SIM. This asset is originally described in: [6].

ISO 15118 Response Codes (I,A,P) Most response messages sent during ISO 15118 communication contain a response code which carries information on the status of the request. Might be stored on the CP inside logs (for possible error checking / maintainence). This asset is also referred to as: ResponseCode [5]. This asset is originally described in: [4, 5].

ISO15118 Session Identifier (C,I,A,P) Identifies an ISO15118 session and is part of the header of every message sent during the communication. It is first established during session setup as part of the SessionSetupRes message. It is also explicitly send during the MeterinReceiptReq to bind the signed metering values to a specific session. If the EVCC wants to resume a previous session it needs to store this value together with the chosen SelectedPaymentOption and RequestedEnergyTransferMode. This asset is also referred to as: SessionID [5]. This asset is originally described in: [4, 5].

ISO 15118 Version Number (I,A,P) Encodes the current ISO 15118 version number using the VersionNumberMajor and VersionNumberMinor parameters. Transmitted during supportedAppProtocolReq but also stored on the EV. The CP will also store the supported app versions, even though it might be part of its protocol implementation and thus firmware. Same holds for the EV. This asset is also referred to as: VersionNumberMajor [5], VersionNumberMinor [5]. This asset is originally described in: [4, 5].

List of Root Certificate IDs (C,I,A,P) Contains a list of known Root Certificate IDs. Used during CertificateUpdateReq by the EV to indicate which V2G Root and MO Root CA certificates are known by the EV and can thus be used to generate a new Contract Certificate. This asset is also referred to as: ListOfRootCertificateIDs [5]. This asset is originally described in: [4, 5].

Meter Identifier (C,I,A,P) Uniqely identifes the meter of a CP. Sent during MeteringReceiptReq. This asset is also referred to as: MeterID [5]. This asset is originally described in: [4, 5].

Meter Information Data Tuple (C,I,A,P) Contains the aggregated data of a Meter. Used during the MeteringReceiptReq, ChargingStatusRes, and CurrentDemandRes messages. Contains the MeterID, MeterReading, SigMeterReading, MeterStatus, and TMeter. This asset is also referred to as: MeterInfo [5]. This asset is originally described in: [4, 5].

Meter Reading (C,I,A,P) Encodes the current meter reading in W/h. Part of the MeterInfo data and thus sent as part of the MeterReceiptReq and ChargingStatusRes messages. This asset is also referred to as: MeterReading [5]. This asset is originally described in: [4, 5].

Meter Reading Signature (C,I,A,P) Represents the signature over the meter reading created by the meter inside the CP. Sent as part of the MeterInfo data and thus sent as part of the MeterReceiptReq and ChargingStatusRes messages. This asset is also referred to as: SigMeterReading [5]. This asset is originally described in: [4, 5].

MeterSerialNumber (I,A) Uniquely identifies the electric meter inside a CP. May also be used to identify a CP indirectly. This asset is also referred to as: MeterID [5]. This asset is originally described in: [6].

Meter Status (C,I,A,P) Contains current status information of a meter inside the CP. Transported to the EV in certain requests, but exact format is left out of scope for CPOs to define. This asset is also referred to as: MeterStatus [5]. This asset is originally described in: [4, 5].

Meter Timestamp (C,I,A,P) Encodes the current time on the meter inside the CP using the Unix Time Stamp format. Sent as part of the MeterInfo data and thus sent as part of the MeterReceiptReq and ChargingStatusRes messages. This asset is also referred to as: TMeter [5]. This asset is originally described in: [4, 5].

MeterType (I,A) Depicts information about the meter built into a CP. This data is stored on the CP and can be accessed by the CPO. May be used to identify a CP. This asset is originally described in: [6].

Meter Values (C,I,A) Each CP has at least one electrical meter that will generate metering related data during a charging session. The format of these meter values generated during a charging session may vary between absolute values and start and end values. This asset incorporates all types of meter value representations. This asset is originally described in: [6].

OEM Provisioning Certificate (I,A,P) This certificate is installed in the EV by the OEM during the manufacturing process. It is used during the Certificate Installation Request to verify the signature of the request message. It uniquely identifies the vehicle for its life time. The certificate is stored on the EV, probably the OEM Sub 2 CA that signed it, and also maybe the OEM backend. The certificate contains information on Issuer (Country, Organization, Common Name) and Subject (Organization, Common Name

(PCID), and Domain Component ("OEM")) among other things. This asset is originally described in: [4, 5].

OEM Provisioning Certificate Private Key (C,I,A,P) Private key corresponding to the OEM Provisioning Certificate. Used to sign Certificate Installation Request messages (or at least parts of it) to verify the request came from a valid EV. Stored on the respective EV / EVCC during manufacturing. This asset is originally described in: [4, 5].

Parameter Set Identifier (C,I,A,P) Uniquely identifies a set of parameters that belong to a specific service provided by a CP. One service may support several parameter sets. Transfered as part of the ServiceDetailRes and PaymentServiceSelectionReq. This asset is also referred to as: ParameterSetID [5]. This asset is originally described in: [4, 5].

ParentID (C,I,A,P) Depending on how MO and CPOs handle and implement fleet management, it may be necessary to subsume all contract specific identifiers used for authorization under a so called ParentID which is used for the authorization. The ParentID is a way for CPOs and CPs to group authorizations together even though different other identifiers are used. These might be stored by the CP for authorization and logging as well as by the Central System. This asset is originally described in: [6].

Payment Options (C,I,A,P) Part of the ServiceDiscoveryRes and details the available payment options the EV may chose to pay the requested services. This asset is also referred to as: PaymentOption [5]. This asset is originally described in: [4, 5].

PE SECC Certificate (I,A) This certificate is the equivalent of the SECC Certificate but only used in private environments. It is signed by the PE Private Operater Root CA and is used to authenticate the wallbox / charge point against the EV. It is stored on the SECC of the wallbox / charge point. The certificate contains information on Issuer (Country, Organization, Common Name) and Subject (Organization, Common Name (PEID), Domain Component, and Country) among other things. This asset is originally described in: [4, 5].

PE SECC Certificate Private Key (C,I,A) Private key corresponding to the PE SECC Certificate. Used to sign authenticate the wallbox / charge point against the EV. Typically stored in the SECC or on the wallbox / charge point. This asset is originally described in: [4, 5].

Power Delivery Parameters (C,I,A,P) These parameters are sent by the EVCC as part of the PowerDeliveryReq. They contain the current ChargeProgress, SAScheduleTupleID, ChargingProfile, and DC_EVPowerDeliveryParameters. These are sent by the EV to initiate the power delivery from the CP and are thus probably stored by both entities. Both will probably log this information as well. This asset is originally described in: [4, 5].

Private Environment Identifier (I,A,P) Uniquely identifes a PE SECC Certificate. Pretty much used in the same way as a CPID, but only used in private environments. This asset is also referred to as: PEID [5]. This asset is originally described in: [4, 5].

Provisioning Certificate Identifier (C,I,A,P) Uniquely identifies the OEM Provisioning Certificate of an EV installed during manufacturing. It is communicated as part of the OEM Provisioning Certificate inside the CertificateInstallationReq by the EV as a means for the MO to identfy whether the customer has a valid contract with him. Typically, the MO receives the PCID from the customer during the registration process to bind the contract to a specific EV. The MO will generate contract certificates based on the validity of the contract. This asset is also referred to as: PCID [5]. This asset is originally described in: [4, 5].

Public Diffie Hellman Encryption Key (C,I,A,P) Public key portion of the ECDH key used to encrypt the private key corresponding to a new contract certificate. Sent as part of the CertificateUpdateRes and CertificateInstallationRes messages. While ISO15118-2 assumes the CCH will probably store the contract certificate and private key, VDE AR E 2802-100-1 assumes that the CCP stores them as part of the messages generated in advance. It should be noted that both only propose this as a process for generation and installation but do not define it as the single valid solution. This asset is also referred to as: DHpublickey [5]. This asset is originally described in: [4, 5].

Require Metering Receipt Flag (C,I,A,P) Indicates to the EVCC if it should sent a MeteringReceiptReq with signed metering receipt from the vehicle. Sent as part of the ChargingStatusRes and CurrentDemandRes This asset is also referred to as: ReceiptRequired [5]. This asset is originally described in: [4, 5].

ReservationID (C,I,A,P) Whenever a customer reserves a CP to charge his EV at a certain time period, he will be supplied with a ReservationID. This ID will be stored in the MO backend and will probably be distributed to the CPO and its CP as well. It uniquely identifies the reservation and is also linked to the Customer as well. It may thus be usable to identify a Customer / EV transitively. This asset is originally described in: [6].

SECC Certificate (I,A) This certificate belongs to the charge point, or more specifically, to the SECC of the charge point. It is used by the EV during TLS communication setup to make sure the EV is talking to a real and not a fake charge point. The certificate is stored on the charge point and probably inside the CPO backend. The certificate contains information on Issuer (Country, Organization, Common Name), Subject (Organization, Common Name (CPID), Domain Component ("CPO"), and Country) among other things. This asset is also referred to as: EVSE Leaf Certificate [5]. This asset is originally described in: [4, 5].

SECC Certificate Private Key (C,I,A) Private key that corresponds to the SECC Certificate. The key is used during TLS connection establishmend when the EV sends a challenge to the charge point. The key is stored on the charge point. This asset is originally described in: [4, 5].

Secondary Actor Schedule Tuple Identifer (C,I,A,P) The SAScheduleTupleID is used to uniquely identify the SAScheduleTuple used during the ISO15118 charging session. It is stored on the EVCC in case the EV wants to resume a previous connection. This asset is also referred to as: SAID [5]. This asset is originally described in: [4, 5].

Service (I,A) Datatype that contains all the information about a service offered by a CP. This data type contains data like the ServiceID, Name, Scope and Cost. Can also contain additional data like which EnergyTransferMode is supported etc. Is transfered part of the ServiceDiscoveryReq message. This asset is originally described in: [4, 5].

Service Identifier (C,I,A,P) Uniquely identifies a service that is provided by the CP. This value is probably only locally unique per CP but could also be globally unique per CPO. Is transfered during ServiceDiscoveryRes, ServiceDetailReq, ServiceDetailRes, and PaymentServiceSelectionReq. This asset is also referred to as: ServiceID [5]. This asset is originally described in: [4, 5].

Service List (C,I,A,P) List of services that a charge point supports, like VAS for example. Each service has a ServiceID, ServiceName, ServiceCategory, ServiceScope, and FreeService. The later denotes if a service is considered free of charge. This information is transfered during the ServiceDiscoveryRes but also stored (probably in a different format) on the CP as well. This asset is also referred to as: ServiceList [5], ServiceType [5]. This asset is originally described in: [4, 5].

Service Parameters (C,I,A,P) Each service offered by a CP may contain /require a number of parameters that are required by the EV (and the CP) for operation. Typically transferred as response to the ServiceDetailRes message. This information must be stored on the CP as well. This asset is also referred to as: ServiceParameterList [5]. This asset is originally described in: [4, 5].

TransactionID (C,I,A,P) Charging sessions and other services will be authorized from the backend. The CPO and CP will store and identify each authroized transaction by transactionIDs. These can be used to uniquely identify a specific transaction and is thus also linked to a customer. This asset is originally described in: [6].

VendorID (I,A) Uniquely Identifiers the Vendor / Manufacturer of a CP. This asset is also referred to as: ChargePointVendor [6]. This asset is originally described in: [6].

3.1.2 TOE-related Assets

This section lists the assets related to the TOE.

Clock (I,A) Data object representing the SecMod time value that increments each millisecond that the SecMod is powered. A non-volatile value (NV Clock) is updated periodically from Clock.

Context (I,A) Contexts are applicable to objects (User keys and Primary keys) and also sessions (authorizations and sequence).

Credentials (C,I,A) Data objects containing encrypted credential information and the encryption key. It reflects the credential distribution for a key on a SecMod.

Endorsement Hierarchy (I,A) Set of services to manage Privacy Administrator controls.

Endorsement Primary Key (C,I,A) A root key created by the SecMod that may be stored in the SecMod or securely cached outside the SecMod. The resource is instantiated in the Endorsement Hierarchy

Null Hierarchy (I,A) Set of services provided to user World.

Non-volatile (NV) Storage (C,I,A) Non-volatile storage of the SecMod provided to the user and protected by access rights managed by the SecMod owner.

Only-extend Register (I,A) Only-extend register intended to record measurement digests and to be used for attestation and access control.

Platform Hierarchy (I,A) Set of services to manage platform firmware controls.

Platform Primary Key (C,I,A) A root key created by the SecMod that may be stored in the SecMod or securely cached outside the SecMod. The resource is instantiated in the Platform Hierarchy

Random Number Generator (C,I,A) Random Number Generator creates uniform random numbers provided to the user and for internal use(e. g., key, secret, and nonce generation

Storage Hierarchy (I,A) Set of services to manage Owner controls.

Storage Primary Key (C,I,A) A root key created by the SecMod that may be stored in the SecMod or chached outside the SecMod. The resource is instantiated in the Storage Hierarchy

User Key (C,I,A) Any cryptographic key except the primary keys, e.g., ordinary or derived keys.

3.2 Threats

This section lists the threats that are to be mitigated by the TOE, its development environment, its operational environment, or a combination of these three. The threat model considers two different kinds of attackers to the EVCC/SECC and its integrated TOE, distinguishing between their different attack paths: The *local attacker* has physical access to the EVCC/SECC and its integrated TOE or she has established a connection to these components and the *remote attacker* who is located in the Internet and uses the Wide Area Network (WAN) connection for his attack.

AbuseFunctionality *Abuse of functionality.* An attacker with high attack potential tries to use functions of the TOE which shall not be used in TOE operational phase in order (i) to disclose or manipulate sensitive user data or TOE security functionality, (ii) to manipulate the TOE's firmware or (iii) to manipulate (explore, bypass, deactivate or change) security features or functions of the TOE. In particular, the TOE shall ensure that functionality that shall not be usable in the operational phase, but which is present during the phases of the TOE's manufacturing and initialization as well as during the integration phase of the EVCC or SECC and the TOE, is deactivated before the TOE enters the operational phase. Such functionality includes in particular testing, debugging, and initialization functions.

CompromiseInternalData *Compromise of confidential user data or TOE security functionality data.* An attacker with high attack potential tries to compromise confidential user data or TOE security functionality data via the regular communication interface of the TOE. This threat comprises several attack scenarios of revealing confidential internal user data or TOE security functionality data. The attacker may try to compromise the user authentication data, to reconstruct a private signing key by using the regular command interface and the related response codes, or to compromise generated shared secret values or ephemeral keys.

DataScavenging *Scavenging of TOE's residue data.* A user may obtain information that she is not authorized to have when the data in shielded locations is no longer actively managed by the TOE.

DenyAccountability *Deny Accountability of Data.* An originator of data may deny originating the data to avoid accountability.

ExploitCommunication *Eavesdrop, inject, manipulate, and suppress Messages.* An attacker with high attack potential tries to manipulate the communication between the TOE and the EVCC or SECC to disclose, to forge or to delete transmitted (sensitive) data or to insert data in the data exchange. Further, she can capture and replay messages. This threat comprises several attack scenarios. An attacker may read data during data transmission in order to gain access to user authentication data or sensitive material as generated ephemeral keys or shared secret values. An attacker may try to forge public keys during import to (respective export from) the TOE.

ExploitLeakage *Exploitation of unintentional Information Leakage.* An attacker with high attack potential tries to launch an attack against the implementation of the cryptographic algorithms. This threat comprises several attack scenarios. An attacker may try to predict the output of the random number generator in order to get information about a generated session key, shared secret value or ephemeral key. An attacker may try to exploit unintentional side-effects (leakage) that are emitted during the operation of a cryptographic operation in order to compromise the processed keys, the authentication data or to get knowledge of other sensitive TOE security functionality or user data. The sources for this leakage information can be the execution time, the electromagnetic emission, or the power consumption.

ExportManipulation *Export manipulated data.* A user or an attacker may export data from shielded locations without security attributes or with insecure security attributes, causing the data exported to be erroneous and unusable, to allow erroneous data to be added or substituted for the original data, and/or to reveal secrets.

ForgeInternalData *Forgery of user data or TOE security functionality data.* An attacker with high attack potential tries to forge internal user data or TOE security functionality data via the regular communication interface of the TOE. This threat comprises several attack scenarios of forgery of internal user data or TOE security functionality data. The attacker may try to alter user data, e. g., by deleting and replacing persistently stored key objects or adding data to data already stored in elementary files. The attacker may misuse the TOE security functionality management function to change the user authentication data to a known value.

ImportManipulation *Import manipulated data.* A user or attacker may import data without security attributes or with erroneous security attributes, causing key ownership and authorization to be uncertain or erroneous and the system to malfunction or operate in an insecure manner.

Malfunction *Provoke Malfunctioning of the TOE.* An attacker with high attack potential tries to cause a malfunction of the TOE security functionality or of the TOE's firmware by applying environmental stress in order to (i) deactivate or modify security features or functions of the TOE or (ii) circumvent, deactivate, or modify security functions of the firmware. This may be achieved, e. g., by operating the TOE outside the normal operating conditions, exploiting errors in the firmware or misuse of administration function. To exploit this an attacker needs information about the functional operation.

Misuse *Misuse of TOE functions.* An attacker with high attack potential tries to use the TOE functions to gain access to access control protected assets without knowledge of user authentication data or any implicit authorisation. This threat comprises several attack scenarios. The attacker may try to circumvent the user authentication mechanism to access assets or functionality of the TOE that underlie the TOE's access control and require user authentication. For instance, this could be used to impersonate a legitimate user. The attacker may try to alter the TOE security functionality data, e. g., to extend the user rights after successful authentication.

ObjSecAttrManipulation *Manipulation of an Object's Security Attributes.* A user or attacker may create an object with no security attributes or make unauthorized changes to security attribute values for an object to enable attacks.

PhysicalTampering *Physical tampering.* An attacker with high attack potential tries to manipulate the TOE through physical tampering, probing, or modification in order to extract or alter user data or TOE security functionality data stored in or processed by the TOE. Alternatively, the attacker tries to change TOE functions (as, e. g., cryptographic functions provided by the TOE) by physical means (e. g., through fault injection).

3.3 Organisational Security Policies

This section specifies the organisational security policies that the TOE and its environment shall comply with in order to support the EVCC/SECC.

ContextManagement *Control secure Caching.* A resource manager shall be able to secure caching of resources without the knowledge or assistance from the application that loaded the resource.

CryptographicMechnisms *Cryptographic Mechanisms.* The cryptographic mechanisms which are provided by the TOE shall be used according to [1].

DAA *Direct Anonymous Attestation.* The DAA issuer and the TOE owner establish a procedure for attestation without revealing the attestation information (i.e., the identity of the TOE).

Locality *Assertion of Locality.* The EVCC and SECC trusted processes assert their locality to the TOE. The TOE guards access to resources, PCRs, and secure storage, to keys and data to be imported, and to defined commands depending on the execution environment's privilege level.

PolicyAuthorization *Policy Authorization.* The TOE supports multiple trusted processes obeying the principle of least privilege by means of role based administration and separation of duty by configuring policy authorization to allow individual entities (trusted processes, specific privileges, operations).

Random *Random Number Generation.* The TOE shall generate random numbers for its own use (e.g. for the generation of ECC key pairs and session keys) and for use by the EVCC or SECC itself according to [1].

RTMeasurement *Measurements of the Root of Trust for Measurement.* The RTM calculates digests as cryptographic hash values of a representation of embedded data or program code (measured values) for reporting.

RTReporting *Reporting of the Root of Trust for Reporting.* The RTR reports on the contents of the root of trust for storage. A RTR report is typically a digitally signed digest of the contents of selected values within a TOE (measurement, key properties or audit digest). The authenticity of the assets reported is based on the verification of the signature and the certificate of the signing key.

RTStorage *Storing Assets with the Root of Trust for Storage.* The Root of Trust for Storage (RTS) protects the assets entrusted to the TOE in confidentiality and integrity.

3.4 Assumptions

In the following assumptions about the environment of the TOE that need to be taken into account in order to ensure a secure operation of the TOE are listed.

Administration *Administration of the TOE.* The administration of the integrated TOE, in particular related to the administration of the TOE's file and object system consisting of folders, data files, and key objects, takes place under the control of the EVCC/SECC Administrator. The EVCC/SECC Administrator is responsible for the key management on the integrated TOE and takes in particular care for consistency of key material in key objects and associated certificates.

Integration *Integration phase of the EVCC/SECC and TOE.* It is assumed that appropriate technical and/or organisational security measures in the phase of the integration of the EVCC/SECC and the TOE in the TOE life cycle model guarantee for the confidentiality, integrity, and authenticity of the assets of the TOE to be protected with respect to their protection need. In particular, this holds for the generation, installation, and import of initial keys, certificates, and authentication material. The Integrator in particular takes care for consistency of key material in key objects and associated certificates as far as handled in the framework of the integration of the EVCC/SECC and the TOE.

OperationalPhase *Operational phase of the integrated EVCC/SECC.* It is assumed that appropriate technical and/or organisational measures in the operational phase of the integrated EVCC/SECC guarantee for the confidentiality, integrity, and authenticity of the assets of the TOE to be protected with respect to their protection need. In particular, this holds for key, and authentication objects stored, generated, and processed in the operational phase of the integrated EVCC/SECC.

PhysicalProtection *Physical protection of the TOE.* It is assumed that the TOE is physically and logically embedded into a EVCC/SECC that is certified according to its relevant protection profile (whereby the integration is performed during the integration phase of the life cycle model). It is further assumed that the EVCC/SECC is installed in a non-public environment within the premises of the consumer that provides a basic level of physical protection. This protection covers the EVCC/SECC, the TOE, the peripherals that the EVCC/SECC communicates with and the communication channel between the EVCC/SECC and the TOE.

TrustedAdmin *Trustworthiness of the EVCC/SECC Administrator.* It is assumed that the EVCC/SECC Administrator is trustworthy and well-trained, in particular in view of the correct and secure usage of the TOE.

4 Security Objectives

4.1 Security Objectives for the TOE

The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The TOE provides security functionality to solve a certain part of the problem defined by the security problem definition. This part wise solution is called the security objectives for the TOE and consists of a set of statements describing the security goals that the TOE should achieve in order to solve its part of the problem.

O.AbuseFunctionality Protection against abuse of functionality. The TOE shall prevent that functions intended for the testing and production of the TOE and which must not be accessible after TOE delivery can be abused in order (i) to disclose or manipulate sensitive user data or TOE Security Function (TSF) Data, (ii) to manipulate the TOE's software or (iii) to bypass, deactivate, change or explore security features or functions of the TOE.

O.AccessControl Access control for functionality and objects. The TOE shall provide and enforce the functionality of access right control. The access right control shall cover the functionality provided by the TOE (including its management functionality) and the objects stored in or processed by the TOE. The TOE shall enforce that only authenticated entities with sufficient access control rights can access restricted objects and services. The access control policy of the TOE shall bind the access control right to an object to authenticated entities.

O.Authentication Authentication of external entities. The TOE shall support the authentication of users as well as to authenticate itself.

O.Confidentiality Confidentiality of user data or TSF Data. The TOE shall ensure the confidentiality of private keys and other confidential user data and confidential TSF Data (especially the user authentication data as the GW-PIN) under the TSF scope of control.

O.ContextManagement The TOE must ensure a secure wrapping of a resource (except seeds) in a manner that securely protects the confidentiality and the integrity of the data of this resource and allows the restoring of the resource on the same TOE and during the same operational cycle only - TOE operational cycle is a Startup Clear to a Shutdown Clear and contexts cannot be reloaded across a different Startup Clear to Shutdown Clear cycle from the one in which they are created.

O.CryptoKeyMan The TOE must manage cryptographic keys, including generation of cryptographic keys using the TOE random number generator as source of randomness, in a manner to protect their confidentiality and integrity.

O.DAC The TOE must control and restrict user access to the TOE protected capabilities and shielded locations in accordance with a specified access control policy where the object owner manages the access rights for their data objects using the principle of least privilege.

O.ECDAA The TOE must support the TOE owner for attestation to the authenticity of measurement digests without revealing the attestation information by implementation of the TOE part of the ECDAA.

O.Export When data are exported outside the TOE, the TOE must securely protect the confidentiality and the integrity of the data as defined for the protected capability. The TOE shall ensure that the data security attributes being exported are unambiguously associated with the data.

O.FailSecure The TOE must enter a secure failure mode in the event of a failure.

O.GeneralIntegChecks The TOE must provide checks on system integrity and user data integrity.

O.Import When data are being imported into the TOE, the TOE must ensure that the data security attributes are being imported with the data and the data is from an authorised source. In addition, the TOE shall verify those security attributes according to the TSF access control rules. TOE supports the protection of confidentiality and the verification of the integrity of imported data.

O.Integrity Integrity of user data or TSF Data. The TOE shall ensure the integrity of the user data, the security services provided by the TOE and the TSF Data under the TSF scope of control.

O.KeyAgreementDH DH key agreement. The TOE shall securely implement the DH key agreement (ECKA-DH) as used in ISO/IEC15118.

O.KeyManagement Key management. The TOE shall enforce the secure generation, import, distribution, access control and destruction of cryptographic keys.

O.Leakage Leakage protection. The TOE shall be designed and built in such a way as to control the production of intelligible emanations within specified limits. The TOE shall provide side channel resistance, i. e., shall be able to prevent appropriately leakage of information, e. g., electrical characteristics like power consumption or electromagnetic emanations that would allow an attacker to learn about private key material, confidential results or intermediate results of cryptographic computations, authentication data.

O.LimitActionsAuth The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user.

O.Locality The TOE must control access to objects based on the locality of the process communicating with the TOE.

O.Malfunction Protection against malfunction of the TOE. The TOE shall ensure its correct operation. The TOE shall prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. The TOE shall preserve a secure state to prevent errors and deactivation of security features of functions. The environmental conditions include external energy (esp. electromagnetic) fields, voltage (on any contacts), clock frequency, and temperature.

O.MessageNR The TOE must provide user data integrity, source authentication, and the basis for source non-repudiation when exchanging data with a remote system.

O.NoResidualInfo The TOE must ensure there is no "object reuse", i. e., there is no residual information in information containers or system resources upon their reallocation to different users.

O.RecordMeasurement The TOE must support calculating hash values and recording the result of a measurement.

O.Reporting The TOE must report measurement digests and attest to the authenticity of measurement digests.

O.SecurityAttrMgt The TOE must allow only authorized users to initialize and to change security attributes of objects and subjects. The management of security attributes shall support the principle of least privilege by means of role based administration and separation of duty.

O.SecurityRoles The TOE must maintain security-relevant roles association of users with those roles.

O.SelfTest The TOE must provide the ability to test itself, verify the integrity of the shielded data objects and the protected capabilities operate as designed and enter a secure state in case of detected errors.

O.Sessions The TOE must provide the confidentiality of the parameters of the commands within an authorised session and the integrity of the audit log of the commands.

O.Sign Signature generation and verification. The TOE shall securely generate and verify digital signatures as used in ISO/IEC15118.

O.SingleAuth The TOE must provide a single user authentication mechanism and require re-authentication to prevent “replay” and “man-in-the-middle” attacks.

O.TamperResistance The TOE must resist physical tampering of the TSF by hostile users. The TOE must protect assets against leakage.

O.TrustedChannel Trusted channel. The TOE shall establish a trusted channel for protection of the confidentiality and the integrity of the transmitted data between the TOE and the successfully authenticated users. The TOE shall enforce the use of a trusted channel if defined by the access condition of an object.

4.2 Security Objectives for the Operational Environment

The following defines the security objectives for the operational environment of the TOE.

OE.Configuration The TOE must be installed and configured properly for starting up the TOE in a secure state. The security attributes of subjects and objects shall be managed securely by the authorised user.

OE.Credential The IT environment must create EK and AK credentials by trustworthy procedures for the root of trust for reporting.

OE.ECDAA The ECDAA issuer must support a procedure for attestation without revealing the attestation information based on the ECDAA signing operation.

OE.Locality The developer of the host platform must ensure that trusted processes indicate their correct locality to the TOE and untrusted processes are only able to assert the locality 0 to the TOE.

OE.Measurement The platform part of the root of trust for measurement provides a representation of embedded data or program code (measured values) to the TOE for measurement.

OE.PhysicalProtection Physical protection of the TOE. The TOE shall be physically and logically embedded into a EV or CP.

OE.TrustedChannel Trusted channel. The users shall establish a trusted channel between the users and the TOE for protection of the confidentiality and integrity of the sensitive data transmitted between the authenticated user and the TOE.

4.3 Rationale

The following provides information on how threats are mitigated by the security objectives for the TOE and the operational environment.

AbuseFunctionality This threat is countered by the security objective O.AbuseFunctionality, which averts the threat by ensuring that functions intended for the testing and/or production of the TOE are not accessible after TOE delivery.

ForgeInternalData This threat is countered by the security objective O.Integrity, which manages the integrity of the User and TSF Data under the TSF scope of control as well as for the integrity of the security services provided by the TOE.

CompromiseInternalData This threat is countered by the security objective O.Confidentiality, which manages the confidentiality of the User and TSF Data under the TSF scope of control.

Misuse The security objective O.AccessControl counters this threat by enforcing an access control policy defined for the TOE. The security objective O.Authentication further regulates the access to the TOE's functionality and assets stored and processed by the TOE. The security objectives O.Integrity and O.Confidentiality ensure the protection of the assets independent of the TOE functionality.

PhysicalTampering This threat is countered by the security objective O.TamperResistance. It enables the detection and prevention of physical tampering, probing and manipulation, e. g., by providing resistance to physical tampering. Furthermore, the security objective O.AbuseFunctionality averts the threat by ensuring that functions intended for the testing and/or production of the TOE are not accessible after TOE delivery, which otherwise may be used to enable or enhance physical tampering attacks.

Malfunction This threat is countered by the security objective O.Malfunction, which averts the threat by ensuring the TOE's correct operation and preservation of a secure state, even under abnormal environmental conditions.

5 Security Requirements

5.1 Overview

This part of the draft PP defines security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the Functional Security Requirements (SFRs), which are to be satisfied by the TOE. The requirements comprise functional components from CC Part 2 [3]. On the component level, the CC allows several operations on these security requirements: refinement, selection, assignment and iteration as defined in CC Part 1 [2]. The following sections list the requirements and describe the planned operations for a full protection profile.

5.2 Communication (Class FCO)

The TOE supports the attestation protocol to selectively proof internal data (PCR values) to subjects (cf. Section 1.4.3).

FCO_NRO.1 (Selective proof of origin) Selective proof of origin, requires the TSF to provide subjects with the capability to request evidence of the origin of information. This functionality must be iterated for measurement, reporting, and credentials.

5.3 Cryptographic Support (Class FCS)

As described in Section 1.4.3, the SecMod provides cryptographic services for the EVCC/SECC. Accordingly, multiple functionalities from the cryptographic support class (FCS) are applicable.

FCS_COP.1 (Cryptographic operation) Cryptographic operations must be performed. Note, that this functionality has to be iterated as well.

FCS_CKM.1 (Cryptographic key generation) Cryptographic key material must be generated. Note, that this functionality has to be iterated for all the ciphers (e.g. ECDSA, HMAC, etc.).

FCS_CKM.4 (Cryptographic key destruction) Cryptographic key material must be destroyed so that it cannot be recovered.

FCS_RNG (Random number generation) A deterministic as well as a hybrid Random Number Generator (RNG) is required for many operations. The PP must include an extended component definition of such a RNG, called FCS_RNG here.

5.4 User Data Protection (Class FDP)

The TOE stores several user data that must be protected regarding access, export, import, and internal transfer.

FDP_ACC.1 (Subset access control) The TOE must enforce policies that cover a well-defined set of operations on some subset of the objects. At least, this has to be fulfilled for access control, credentials, export and import, internal object hierarchy (e. g., key objects), measurement, reporting, and non-volatile memory.

FDP_ACC.2 (Complete access control) The TOE must enforce policies that cover all possible operations on objects that are included in the Security Function Policies (SFP).

FDP_ACF.1 (Security attribute based access control) The TOE must support an access control mechanism based on security attributes that enforces specified access rules. At least, this has to be fulfilled for access control, credentials, export and import, internal object hierarchy (e. g., key objects), measurement, reporting, non-volatile memory, and operational states.

FDP_ETC.1 (Export of user data without security attributes) The TOE must support the export of user data without the export of its security attributes.

FDP_ETC.2 (Export of user data with security attributes) The TOE must support the export of user data together with its associated security attributes.

FDP_ITC.1 (Import of user data without security attributes) The TOE must support the import of user data ignoring its security attributes.

FDP_ITC.2 (Import of user data with security attributes) The TOE must support the import of user data and its associated security attributes. The association must be ensured.

FDP_ITT.1 (Basic internal transfer protection) The TSF shall protect the TSF data from disclosure when it is transmitted between separate parts of the TOE.

FDP_RIP.1 (Subset residual information protection) The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to and the deallocation of the resource from internal stored and ephemeral secret material like cryptographic keys or credentials.

FDP_SDI.1 (Stored data integrity monitoring) The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors on all objects, based on the user data attributes.

FDP_SDI.2 (Stored data integrity monitoring and action) The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors on all objects, based on the user data attributes. Upon detection of a data integrity error, the TSF shall not use the data and stop the corresponding process accessing the data, warn the entity connected.

FDP_UCT.1 (Basic data exchange confidentiality) The user data must be protected from unauthorised disclosure when they are transmitted or received in the case of an ex- or import. Further, the data that is transmitted to/from the access control mechanism must be protected as well.

FDP_UIT.1 (Data exchange integrity) The TOE must support the detection of modification, deletion, insertion, and replay of data for a firmware update and the detection of modification of imported and exported data.

5.5 Identification and Authentication (Class FIA)

This class addresses the requirements for functions to establish and verify a claimed user identity. Identification and authentication is required to ensure that users are associated with the proper security attributes (e. g., identity, groups, roles, security or integrity levels). The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies.

FIA_AFL.1 (Authentication failure handling) The TOE must take actions when the number of un-/successful authentication attempts meets or surpasses a defined number. This must be iterated over all authentication events.

FIA_ATD.1 (User attribute definition) If the TOE handles individual users it must maintain security attributes, like authentication states, that belong to a user.

FIA_UAU.1 (Timing of authentication) The TSF shall not allow sensitive actions on behalf of the user to be performed before the user is authenticated and shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.4 (Single-use authentication mechanisms) The TSF shall prevent reuse of authentication data related to for single-use authentication mechanisms.

FIA_UAU.5 (Multiple authentication mechanisms) The TSF shall provide Password-based authentication mechanisms, HMAC-based authentication mechanisms, and Policy-based authentication mechanisms to support user authentication. Further it shall authenticate any user's claimed identity according to the specified rules, e. g., if physical presence of a user must be verified.

FIA_UAU.6 (Re-authenticating) The TSF shall re-authenticate the user under the conditions that multiple commands need to be executed in one authorisation session.

FIA_UID.1 (Timing of identification) The TSF shall allow not sensitive actions on behalf of the user to be performed before the user is identified and require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1 (User-subject binding) The TSF shall associate user security attributes with subjects acting on behalf of this user. Further it shall enforce specified rules on the initial association and shall govern changes to the user security attributes.

5.6 Security Management (Class FMT)

This class is intended to specify the management of several aspects of the TSF: security attributes, TSF data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

FMT_MOF.1 (Management of security functions behaviour) The TSF shall restrict the ability to disable and enable the function to clear authorization values to authorised subjects.

FMT_LIM.1 (Limited capabilities) The TSF shall enforce the limitation of capabilities if the TOE is in a specific life cycle phase. For instance, debug or deploying test functions should not be available after the deployment. Otherwise user or TSF data could be disclosed or manipulated and software could be reconstructed and substantial information could be gathered to enable attacks.

FMT_LIM.2 (Limited availability) The availability of the TSF shall be limited if the TOE is in a specific life cycle phase. For instance, debug or deploying test functions should be removed entirely after the deployment. Otherwise user or TSF data could be disclosed or manipulated and software could be reconstructed and substantial information could be gathered to enable attacks.

FMT_MSA.1 (Management of security attributes) The TSF shall enforce the access control mechanism and the information flow control to restrict the ability at least to change default, query, modify, and delete, the security attributes to the authorised identified user or roles. At least, this has to be fulfilled for access control, credentials, export and import, internal object hierarchy (e. g., key objects), measurement, reporting, non-volatile memory, and operational states.

FMT_MSA.2 (Secure security attributes) The TSF shall ensure that only secure values are accepted for the security attributes.

FMT_MSA.3 (Static attribute initialisation) The TSF shall enforce the access control mechanism and the information flow control to provide restrictive default values for security attributes that are used to enforce the SFP and shall allow the authorised identified users and roles to specify alternative initial values to override the default values when an object or information is created. At least, this has to be fulfilled for access control, credentials, export and import, internal object hierarchy (e. g., key objects), measurement, reporting, non-volatile memory, and operational states.

FMT_MSA.4 (Security attribute value inheritance) The TSF shall use specified rules to set the value of security attributes. At least, this has to be fulfilled for user authentication, internal object hierarchy, and non-volatile memory.

FMT_MTD.1 (Management of TSF data) The TSF shall restrict the ability to change default or modify the specific TSF data, like authentication data or policies, to the authorised identified users and roles. At least, this has to be fulfilled for user authentication and non-volatile memory.

FMT_SMF.1 (Specification of Management Functions) The TSF shall be capable of performing the management functions. At least this should be applied to functions related to cryptographic key life cycle, object hierarchies, authorization values, security attributes, TOE life cycle, and attack mitigation techniques.

FMT_SMR.1 (Security roles) The TSF shall maintain the authorised identified roles for general usage and administrative tasks. Further, it shall be able to associate users with roles.

5.7 Protection of the TSF (Class FPT)

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data.

FPT_FLS.1 (Failure with preservation of secure state) The TSF shall preserve a secure state by entering the fail state when failures occur. At least this shall be the case when the TOE fails to restore the last saved state after a restart or resume, exposure to operating conditions, detection of physical manipulation, insufficient entropy on random number generation, detection of failed self-test, errors during processing cryptographic operations, and errors during evaluation of access control rules.

FPT_EMS.1 (TOE emanation) The TOE shall prevent attacks against the secret data, like cryptographic keys, where the attack is based on external observable physical phenomena of the TOE. Such attacks may be observable at the interfaces of the TOE or may be originated from internal operation of the TOE or may be caused by an attacker that varies the physical environment under which the TOE operates. The set of measurable physical phenomena is influenced by the technology employed to implement the security module.

FPT_ITT.1 (Basic internal TSF data transfer protection) The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE.

FPT_PHP.3 (Resistance to physical attack) The TSF shall resist physical manipulation and physical probing to the all TOE components implementing the TSF by responding automatically such that the SFRs are always enforced.

FPT_STM.1 (Reliable time stamps) The TSF shall be able to provide reliable time stamps as number of milliseconds the TOE has been powered since initialisation of the Clock value.

FPT_TST.1 (TSF testing) The TSF shall run a suite of self tests during initial start-up, periodically during normal operation to demonstrate the correct operation of the TSF, and on request.

5.8 Trusted path/channels (Class FTP)

Families in this class provide requirements for a trusted communication path between users and the TSF, and for a trusted communication channel between the TSF and other trusted IT products. A trusted channel is a communication channel that may be initiated by either side of the channel, and provides non-repudiation characteristics with respect to the identity of the sides of the channel.

FTP_ITC.1 (Inter-TSF trusted channel) The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. The TSF shall permit another trusted IT product to initiate communication via the trusted channel. The TSF shall initiate enforce communication via the trusted channel for any data exchange between the TOE and the EVCC/SECC except reading out the data fields with technical information.

Bibliography

[1] *BSI TR-02102-1 – Kryptographische Verfahren: Empfehlungen und Schlüssellängen*. 2019-01. Bundesamt für Sicherheit in der Informationstechnik. 2019 (cit. on pp. 33, 34).

[2] *Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model*. Version 3.1, Revision 5. Common Criteria for Information Technology Security Evaluation. 2017 (cit. on p. 41).

[3] *Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components*. Version 3.1, Revision 3, Final. Common Criteria for Information Technology Security Evaluation. 2009 (cit. on p. 41).

[4] *ISO/IEC 15118-1:2013, Road vehicles – Vehicle to grid communication interface – Part 1: General information and use-case definition*. 2013. URL: `https://www.iso.org/standard/55365.html` (cit. on pp. 3, 5–8, 10–12, 18–20, 22–29).

[5] *ISO/IEC 15118-2:2014, Road vehicles – Vehicle to grid communication interface – Part 2: Network and application protocol requirements*. 2014. URL: `https://www.iso.org/standard/55366.html` (cit. on pp. 5–8, 10, 12, 18–20, 22–29).

[6] *Open Charge Point Protocol 1.6*. Version 1.6. Open Charge Alliance. 2016. URL: `http://www.openchargealliance.org/protocols/ocpp/ocpp-16/` (cit. on pp. 7, 10, 19–22, 24–30).

[7] *Open InterCharge Protocol for Emobility Service Provider*. Version 2.1. Hubject GmbH. 2016. URL: `https://www.hubject.com` (cit. on pp. 3, 5, 8, 18, 19, 22, 24, 25).

[8] *VDE-AR-E 2802-100-1 - Handling of certificates for electric vehicles, charging infrastructure and backend systems within the framework of ISO 15118*. Version 1. Deutsche Kommission Elektrotechnik Elektronik Informationstechnik (DKE). 2017. URL: `https://www.vde-verlag.de/standards/0800432/vde-ar-e-2802-100-1-anwendungsregel-2017-10.html` (cit. on pp. 4–6, 8, 20).