Fraunhofer SIT

FRAUNHOFER INSTITUTE FOR SECURE INFORMATION TECHNOLOGY

System Security Mechanisms for Electric Vehicles and Charge Points Supporting ISO 15118

Proposal for a Technical Guideline

ATHENE
National Research Center for Applied Cybersecurity

Fraunhofer SIT

System Security Mechanisms for Electric Vehicles and Charge Points Supporting ISO 15118

Proposal for a Technical Guideline

Dustin Kern
Christoph Krauß
Maria Zhdanova
Department Cyber-Physical Systems Security (CSS)
Fraunhofer Institute for Secure Information Technology SIT

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

System Security Mechanisms for Electric Vehicles and Charge Points Supporting ISO 15118

Proposal for a Technical Guideline

Version 1.0, November 2019

Authors:

Dustin Kern
Christoph Krauß
Maria Zhdanova

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-2019-04: System Security Mechanisms for Electric Vehicles and Charge Points Supporting ISO 15118
Dustin Kern, Christoph Krauß, Maria Zhdanova

Contents

1 Introduction

This technical guideline provides recommendations for the secure operation of an e-mobility charging infrastructure. The focus is on system security of the Electric Vehicle (EV) and Charge Point (CP) / Electric Vehicle Supply Equipment (EVSE), with their respective communication control units, the Electric Vehicle Communication Controller (EVCC) and the Supply Equipment Communication Controller (SECC), as well as the secure usage of their communication protocols. Both systems are required to be equipped with a Hardware Security Module (HSM), providing a hardware trust anchor for secure storage and usage of their corresponding private credentials. The trust anchor is also used to provide more advanced security features like software integrity validation or secure firmware updates. Additional recommendations are given, aiming to increase the security of the communication between EVCC and SECC using ISO 15118 [10] as well as the backend communication of the SECC.

1.1 Structure

The remainder of this technical guideline is structured as follows. In Section 2, the scope is defined by describing the considered e-mobility architecture and relevant roles with their functionalities. In addition, the considered threat model is defined and topics which are out-of-scope of this guideline are listed. Section 3 and Section 4 list requirements for securing the EVCC and the SECC respectively. These requirements are based on their ISO 15118 use cases and focus on providing system security (e.g., secure storage for private keys or verifying the integrity of the local software state). Section 5 lists the requirements for an HSM integrated in EVCC and SECC to act as a trust anchor for fulfilling the requirements described in Section 3 and Section 4. In addition, a recommendation is given how the ISO standardized Trusted Platform Module (TPM) 2.0 can be used as HSM meeting the requirements. Section 6 lists requirements for securing the ISO 15118 communication interface between EVCC and SECC with regard to application and transport layer.

1.2 Key Words

The key words “MUST”,“MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [1].

2 Scope

This technical guideline defines technical requirements for providing system security for the EV's EVCC and the CP's SECC and the communication between EVCC and SECC using ISO 15118 [10] as well as the backend communication of the SECC.

This section describes the considered e-mobility architecture in Section 2.1, the relevant roles with their functionalities in Section 2.2, the assumed threat model in Section 2.3, and the topics which are out-of-scope of this guideline in Section 2.4.

2.1 Architecture

Figure 2.1 provides an overview of the e-mobility architecture and the communication links between the different actors, considered as basis for this technical guideline. The EV and CP are each equipped with an HSM, providing the trust anchor needed to match the security requirements of this guideline.

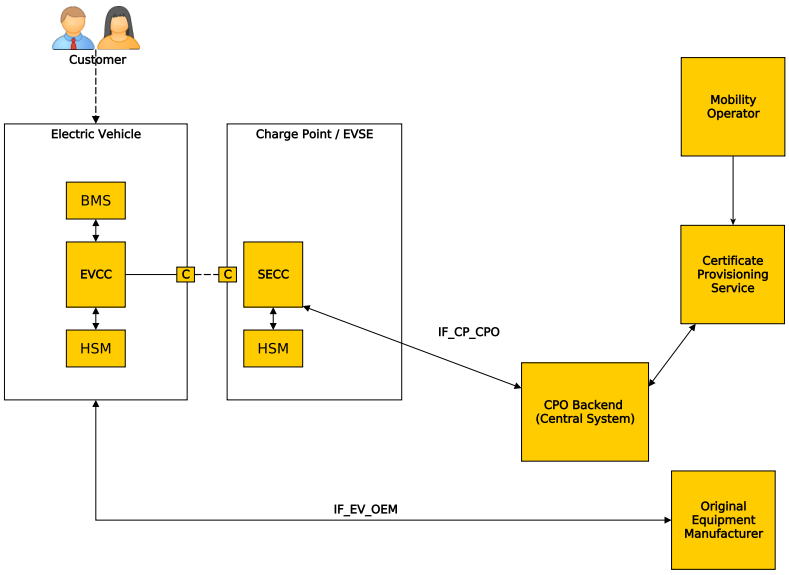

Figure 2.1: E-Mobility Architecture

The EVCC of the EV and the SECC of the CP communicate over the IF_EV_CP interface using Power Line Communication (PLC) with each other. They employ the ISO 15118 protocol [10, 11] to enable Plug-and-Charge (PnC), i.e., automatic

authorization of the charging process with minimal user interaction. For this, the EVCC is authenticated with a signature-based challenge-response mechanism, using its contract credentials (private key and certificate). The EVCC initially receives provisioning credentials from it's Original Equipment Manufacturer (OEM) via the IF_EV_OEM interface. These credentials are later used to request new contract credentials from a Mobility Operator (MO) after the customer (e.g., vehicle owner) has concluded a charging contract with the MO. The requests are sent over the CP, during an ISO 15118 communication session (before charging authorization), and require the EV owner to haven a corresponding contract with the MO. Responses are relayed over a Certificate Provisioning Service (CPS), responsible for validating their correctness and authenticity, and the CP back to the EVCC. For management of the charging process, the EVCC communicates with other EV internal controllers like the Battery Management System (BMS).

The CP's SECC acts as the server in the ISO 15118 communication protocol. ISO 15118 uses TLS 1.2 for confidentiality and integrity protection of its messages, as well as authentication of the SECC. The corresponding CP credentials, i.e., TLS server certificate and respective private key, are installed by it's Charge Point Operator (CPO). Communication between the CP and the CPO backend is handled using OCPP 1.6 [19] over the interface IF_CP_CPO. OCPP 1.6 also relies on TLS for its security with unilateral, server-side authentication. Depending on the OCPP variant, the CP is either authenticated using HTTP Basic in case of OCPP1.6J [20] or a client certificate in case of OCPP1.6S [21].

Both, EVCC and SECC are connected to a HSM, which provides secure storage for critical credentials, e.g., private keys, secure execution environment for critical (cryptographic) operations, and enables additional security functionalities such as measured boot.

2.2 Roles and Actors

The following list summarizes all relevant roles and defines their place within the context of ISO 15118 and e-Mobility. The list is sorted in alphabetical order and does not represent any kind of priority.

Battery Management System The Battery Management System (BMS) is part of the EV and handles the management of the batteries within the EV. It manages both electric and thermal functions and provides communication between the battery system and the other EV controllers [10].

Charge Point Operator The Charge Point Operator (CPO) [11] is the entity that operates and manages CPs. The CPO may also be the manufacturer of the CP, but this task could also be off-loaded to an external CP manufacturer. The on-site maintenance of CPs may also be off-loaded to an external service provider as well. The CPO can also be split into Sub Operator and Hub Operator [9] and is also referred to as EVSE Operator [10]. The role is

also sometimes divided into the business operator and technical operator role [6], 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 Authority (CA)s, which are necessary to generate the EVSE Leaf Certificates for the respective CPs [11].

Central System The Central System (CPO Backend) (CS) describes the back end of the CPO that a CP is connected to and communicates with [19]. The CS manages and configures all CPs of a CPO and is used in case a CP is not able to authenticate a customer (based on the contract certificate) and to authorize a charging process remotely. After a charging process is finished, the CS collects and compiles the measurements to charge the customer directly or to generate a Charge Detail Record (CDR) in case the customer has a contract with an MO other than the respective CPO of the CP. The CS can also gather intermediate meter values as well and may also provide additional services for a customer.

Certificate Provisioning Service The Certificate Provisioning Service (CPS) [11] 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 taken over by a CPO, MO, or a third party provider [11]. 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 E-Mobility Account Identifier (EMAID) from the respective MO to enable the EV to verify the integrity and authenticity of the aforementioned 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 relayed to the EV [11].

Customer The Customer [6] is the entity that has a contract with an MO and wants to charge an EV. While these could be single persons, it could also be a business that owns a fleet of vehicles that are used by it's employees. In both cases the owners are also the owner of these EVs. In case of a charging session, the customer is often also the EV driver, but in case of a fleet, the customer may also provide identifiers to it's employees that are grouped under it's own identifier to build a group [19]. The customer is also referred to as E-Mobility Customer [6] or User [19].

Electric Vehicle The Electric Vehicle (EV) 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 CP [10, 19]. Once the EV is connected to the CP it will try to communicate and charge using the protocol specified in ISO 15118 [11]. The EV is equipped with its own unique OEM Prov Certificate, a Contract Certificate specifically issued to the EV owner, and several 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.

Electric Vehicle Communication Controller The Electric Vehicle Communication Controller (EVCC) is part of the EV and implements the ISO 15118 communication and all necessary utilities [10, 11].

Electric Vehicle Supply Equipment The Electric Vehicle Supply Equipment (EVSE) is a different term to describe a CP for EVs [10, 19]. The EVSE is equipped with a EVSE Leaf Certificate and the respective certificate chain up to an CPO Sub CA-1 which is used to authenticate it to an EV. The EVSE belongs to a specific CPO and is managed by it.

Mobility Operator The Mobility Operator (MO) [11, 6] 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 CP. To charge the EV at the CP, the CP 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 [9] which can be further separated into Sub Providers and Hub Providers [9], or as an E-Mobility Operator [10].

Original Equipment Manufacturer The Original Equipment Manufacturer (OEM) [10, 6] 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 CAs, which are needed to generate the OEM Provisioning Certificate for the EV.

Supply Equipment Communication Controller The Supply Equipment Communication Controller (SECC) belongs to the EVSE and implements the ISO 15118 communication and all necessary utilities [10, 11]. It is also sometimes referred to as Local Controller [19].

V2G Root-CA The Vehicle to Grid (V2G) Root CA is the main trust anchor of the Public Key Infrastructure (PKI) defined by ISO 15118 [11]. The V2G Root-CA is responsible to certify the Sub-CAs for CPOs and CPS Provider and optionally certify MOs and OEMs as well. It is also certifying the Online Certificate Status Protocol (OCSP) Responder for the CPO Sub-CA 1. While in general, a single central V2G Root-CA is preferred by [6], it can be assumed that there are typically several active and valid V2G Root-CAs available at once. These may be used to implement different regulatory requirements for example. EVs are typically equipped with up to five different V2G Root-CA certificates. This role may be assumed by governments, third party suppliers or even groups of CPOs, OEMs, MOs, and Contract Clearing House (CCH)s.

2.3 Threat Model

The threat model addressed by this technical guideline considers a powerful attacker who has a full physical access to the EV and can modify or replace its components, install malicious firmware or outdated firmware with known vulnerabilities, and extract any stored information except when the data is tamper protected, e.g., using a HSM. The attacker is trying to reach one or multiple of the following goals:

- Extract, copy or duplicate the PnC credentials, i.e., the contract certificate or OEM provisioning certificate together with their respective private keys. The attacker can readout, intercept or duplicate the keys during the system usage, deployment or using recorded messages by carrying out: (i) offline attacks against Flash; (ii) online attacks against RAM; (iii) online attacks via firmware exploits or firmware manipulation. Extracting these credentials allows the attacker to gain unauthorized access to the charging infrastructure, impersonate the EV owned by the legitimate customer of the MO, charge other EVs on behalf of this customer, or request new PnC credentials; thus, invalidating the original ones.
- Extract intellectual property rights-related data and/or privacy sensitive information that stays persistent across firmware upgrades (e.g., a CPs authorization cache). By installing manipulated or outdated, exploitable firmware the attacker can readout these data, harming the manufacturers intellectual property rights and/or the users privacy.
- Manipulate the system's firmware or exchange its components in order to change its behavior (e.g., enabling charging without authentication, sending data to unauthorized third parties or providing falsified measurements). Since charging and billing relies on the information provided by EV and CP (e.g., charging control data, physical limits, charge plans, meter information, CDRs), the attacker can abuse this to gain monetary benefits. A manipulated firmware can also provide the attacker with remote access to EVCC/SECC in order to, e.g., prepare a large scale attack on the grid.

2.4 Out-of-Scope

Within this technical guideline, we describe only specific aspects for securing an EVCC of an EV, an SECC of a CP, and their communication. We assume that appropriate additional measures are taken to address common security requirements, e.g., deactivation of default system passwords or enabling only necessary services. Such requirements can be found for example in the PCI DSS specification [22]. In addition, we assume that appropriate mechanisms for cybersecurity risk management are deployed (cf. for example [17]).

The backend of the e-mobility architecture (MO, CPS, and CS) and the EV's OEM are assumed to be secure and appropriate security mechanisms are deployed. Thus, the provided data (e.g., firmware updates provided by the OEM) is considered trustworthy. Communication of the CP with the backend is assumed to use the Open Charge Point Protocol (OCPP). Detailed security requirements for other communication protocols (e.g., how to implement remote attestation) are out-of-scope. Requirements for CP administration (e.g., access control) are out-of-scope.

Changes to the ISO 15118 standard are out-of-scope. The only exception are optional recommendations which extend the standard but can be easily supported (e.g., TLS 1.3 instead of TLS 1.2 in Section 6.3) and the potentially increased size of encrypted contract keys in Section 3.3.2 which is anyway usually not checked. Changes to the e-mobility PKI of ISO 15118 as well as additional requirements regarding certificate revocation are also out-of-scope.

3 Requirements for the EVCC

3.1 Overview

The basic use cases which the requirements in this section are meant to secure are listed in Section 3.2. They mainly consist of the installation of initial credentials on the EVCC by its OEM during manufacturing, the provisioning of new credentials before a charging process, and using the credentials for authentication of a charging session.

Section 3.3 lists requirements for the EVCC regarding the security of these use cases. The requirements ensure secure storage, provisioning, and usage of its credentials, while also ensuring the integrity of the EVCC's software state and providing means for a secure firmware update.

3.2 Use Cases

The requirements for securing the EVCC are centered around its application for ISO 15118 PnC sessions and are based on the following use cases:

- **EV Initialization:** During the manufacturing process, an EVCC receives its provisioning credentials from the OEM. These credentials are later used to request new contract credentials during an ISO 15118 communication session.
- **Contract Provisioning:** For the provisioning of new contract credentials, the EVCC sings a certificate request, that includes its current certificate, using its current private key. This request is sent to the SECC from where it is forwarded to the MO. The MO generates new contract credentials and encrypts the new contract private key using the public key from the EVCC's old certificate. The new credentials are sent to the CPS where they are validated and signed. The signed credentials are transmitted to the SECC from where they are forwarded to the EVCC. The EVCC decrypts its new contract private key and can use its new credentials for authorization of ISO 15118 PnC sessions.
- **Plug and Charge:** If the identification mode PnC is chosen in an ISO 15118 communication session, the EVCC is authenticated via a challenge-response mechanism. Therefore, the SECC sends a 16 byte nonce as a challenge to the EVCC which has to sign the nonce using its private key

of the contract credentials. Charging is authorized after a successful validation of the EVCC's signature. During a charging session, the EVCC's contract credential private key might additionally be used for signing metering receipts.

- **Logging of Events:** During its operation, the EVCC might create and locally store different kinds of logging information. Some of these logs could potentially include sensitive or privacy relevant data (e.g., a log with the timestamps and charge point IDs of all EV charging processes which could be used to create a movement profile of the customer).

3.3 Securing the EVCC

This section describes the requirements for an EVCC regarding the secure storage, provisioning, and usage of its ISO 15118 PnC credentials. To ensure confidentiality and integrity of the EVCC's private keys (used as PnC credentials) throughout their entire life-cycle an HSM shall be used which meets the requirements described in Section 5. In addition, keys shall only be usable if the EVCC is not manipulated. Thus, key use shall be bound to the integrity of the EVCC's local software state.

3.3.1 Key Storage

During an ISO 15118 PnC charging session, the EVCC uses different keys for authentication and confidentiality protection. The ISO 15118 defined EVCC key pairs are:

OEM Provisioning Key: An EC key pair on the secp256r1 curve. It is installed by the EV's OEM during manufacturing. It is used to sign ISO 15118 requests for the installation of new PnC contract credentials (ECDSA-SHA256) and to en-/decrypt the corresponding private contract keys (AES-CBC-128 with session key from ECDH).

Contract Key(s): EC key pairs on the secp256r1 curve. They are generated by the MO and installed using ISO 15118. They are used to sign ISO 15118 requests for the update of new PnC contract credentials (ECDSA-SHA256) and to en-/decrypt the corresponding private contract keys (AES-CBC-128 with session key from ECDH). Additionally, they are used to sign ISO 15118 authorization and metering receipt requests (ECDSA-SHA256).

For protection against side-channel/run-time attacks, the EVCC's private keys used in ISO 15118, MUST be securely stored in an HSM. The AES session keys MAY be used outside of the HSM but MUST be permanently deleted, after they are no longer needed. The SHA256 hashes for the ECDSA signatures MAY be calculated outside of the HSM, but the signatures themselves MUST be calculated within it.

3.3.2 Key Provisioning

New contract keys, resulting from the ISO 15118 provisioning process (either certificate installation or update), MUST be imported into the HSM for secure, long-term storage. For this, the EVCC MUST possess a new EC key pair (secp256r1 curve), in the following referred to as storage key. The storage key MUST be stored in the HSM and its public portion MUST be included as an octet string in the certificate of the OEM provisioning key, in a non-critical X.509v3 extension. This extension MAY be a sequence of multiple fields if more values need to be transmitted. For compatibility reasons all actors SHOULD know and use the same OID for this extension. The currently RECOMMENDED OID for this draft is 1.3.36.15.9.2.1.1 `[iso(1) identified-organization(3) teletrust(36) TeleTrusT Identified Organisation (15) Fraunhofer Institute for Secure Information Technology SIT (9) Security Objects(2) KeyTypes(1) HSM-StorageKey(1)]`. The resulting certificate structure on the example of a provisioning certificate is shown in Table 3.1. Here, extensions are marked with ^{c} for critical or ^{nc} for non-critical.

Table 3.1: Provisioning Certificate with Storage Key Extension

OEM Provisioning Certificate			
Version:		X.509v3 (0x2)	
Serial Number:		12345 (0x3039)	
Signature Algorithm:		ecdsa-with-SHA256	
Issuer:		CN=OEMSubCA2, O=Orga, C=DE	
Validity	Not Before:	May 7 08:40:32 2019 GMT	
	Not After:	May 6 08:40:32 2021 GMT	
Subject:		CN=PCID, O=Orga, DC=OEM	
Subject Public Key Info	Public Key:	OCTET STRING	
	Algorithm:	id-ecPublicKey	
	Parameters:	namedCurve secp256r1	
X509v3 Extensions	Basic Constraints:^{c}	CA:FALSE	
	Key Usage:^{c}	Digital Signature, Key Agreement	
	Subject Key Identifier:^{nc}	keyIdentifier (SHA-1)	
	HSM Extension:^{nc}	**EC Storage Key 512 bit OCTET STRING**	
		OPTIONAL	⋮
Signature	Algorithm:	ecdsa-with-SHA256	
	Value:	OCTET STRING	

The storage key MUST be usable for a direct import of new, encrypted contract keys into the HSM, i.e., decryption is handled internally by the HSM during the

import such that the contract key never leaves the HSM in unencrypted form. Additionally, the storage key MUST NOT be usable for general-purpose decryption of the contract keys, i.e., it SHALL NOT be possible to decrpyt contract keys with the storage key without directly importing them.

An MO that receives an installation request for a new contract key with an OEM provisioning certificate that includes the HSM extension MUST encrypt the new contract key in such a way that enables the requesting EVCC to directly import it into its HSM using the storage key. Any additional info that is needed by an MO to encrypt the contract key for a specific HSM SHOULD also be included in the HSM certificate extension. The MO MUST also include the HSM extension from the received provisioning certificate into the new contract certificate using the same, non-critical X.509v3 extension. This enables the same process for ISO 15118's certificate update requests, where the EVCC sends it's current contract certificate, instead of the provisioning certificate, to the MO. An EVCC that receives an installation response with a contract certificate that includes the HSM extension MUST directly import the new contract key into its HSM using its storage key. The same process MUST be used for certificate updates, i.e., the MO copies the HSM extension of the old contract certificate to the new one and the new contract key is encrypted/imported by the EVCC using the HSM storage key.

If an MO receives a certificate request without the HSM extension they SHOULD still generate a new contract key and encrypt it in the ISO 15118 defined way – i.e., using the provisioning key in case of an installation or the contract key in case of an update – in order to retain backwards compatibility with regard to EVCCs. If an EVCC receives a certificate response without the HSM extension they SHOULD still decrypt the new contract key in the ISO 15118 defined way to retain backwards compatibility with regard to MOs. In the backwards compatibility cases, the EVCC MUST still import the decrypted contract key into its HSM and permanently delete the version outside of the HSM as soon as possible. Table 3.2 provides an overview of the resulting EVCC keys and their usage.

Table 3.2: EVCC Keys and Their Usage

Key	Utilization
OEM Provisioning Key	Signs contract installation request. Optionally decrypts contract keys for backwards compatibility (installation).
Storage Key	Decrypts contract key during direct import (installation & update).
Contract Key(s)	Signs contract update, authorization and metering receipt requests. Optionally decrypts contract keys for backwards compatibility (update).

Note: ISO 15118 restricts the length of the encrypted, contract private key in certificate installation/update responses to a maximum of 48 bytes (cf. [11], Annex C.6). Since encryption might require a specific structure in order to enable a direct import into the HSM, encrypted contract keys possibly exceed this limit. As a result, even in the case where EV and MO support the direct import mechanism, intermediate actors (CPS or CP) might still cause incompatibilities. In this case, the MO MAY use the backwards compatibility mode as a fallback.

The EVCC's keys in the HSM SHALL be saved in a hierarchy such that a subordinate key can only be loaded and used by subsequently loading all keys above it in the hierarchy. Hence, using a key MUST also require the respective authorization for every previous key in the hierarchy. Usage of the storage key MUST NOT be allowed without prior authorization (e.g., password based), in order to prevent unauthorized access by arbitrary services running on the EVCC. Importing a contract key MUST put it below the storage key in the hierarchy such that using a contract key requires prior authorization for the storage key. The OEM provisioning key MUST also be saved underneath the storage key.

Fig. 3.1 shows an example of the resulting key hierarchy. In the squared brackets, information about the respective key's usage is provided for the new storage key as well as for the keys related to the PnC credentials as defined by ISO 15118. The dashed line indicates that the MO does not support encrypting contract keys for a direct import, and hence, the contract key was decrypted on the EVCC prior to being imported it into its HSM.

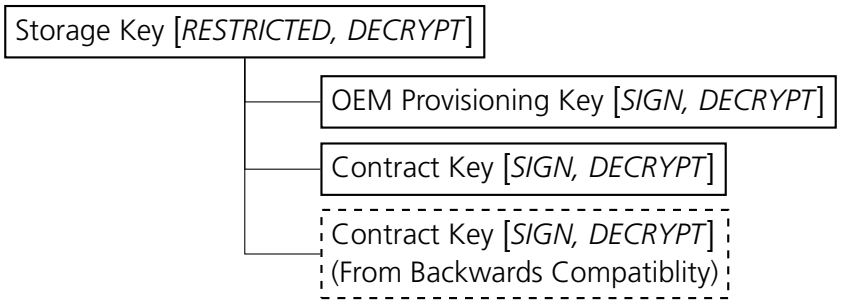

Figure 3.1: HSM Key Hierarchy

Old contract keys SHOULD be deleted from the EVCC's HSM once they are no longer needed, i.e., the corresponding certificate is no longer valid or a new, valid contract key is available.

3.3.3 Software Integrity with Secure Updates

Before every key installation and charging process, the integrity of the EVCC's software MUST be validated. Access to (private) keys shall only be possible if the system is in a trustworthy state.

The private key of the contract credential MUST be bound to a trusted software state, i.e., an authorization policy must be enforced which restricts access to the key stored in the HSM to a list of known trusted software states. This MAY be realized directly (for the actual authorization of the contract key) or indirectly (for the authorization of any key above the contract key in the hierarchy). It is RECOMMENDED to achieve the binding to a trusted software state by:

1. providing a digitally signed, trusted software state alongside every software-version
2. providing the respective public key to validate this signature in an authenticity protected manner to the EVCC
3. binding the use of the contract key (either direct or indirect) to the comparison of any authenticated software state (proven to be authentic by the validation of its signature) to the locally measured state.

Since the contract key is usable with any authenticated software state, this method allows for secure software updates without any changes to the contract key as long as a new, signed state is provided.

To also thwart downgrade attacks, the contract key SHOULD additionally be bound (either direct or indirect) to a comparison of the current firmware version to a monotonically increasing counter maintained by the HSM. At any firmware upgrade, the counter is increased to the new version number and authorization for the contract key requires the current firmware version to be greater or equal to the value of the counter.

Since other EV components, apart from the EVCC, are involved in the charging process (e.g., the BMS), the state of these components SHOULD also be validated before authorization of a charging process. This SHOULD be done by additionally binding the authorization of the contract key (either direct or indirect) to the validation of a signature of all these components. For this, a simple challenge-response protocol MAY be used. If this is the case, the challenge MUST be a random nonce and the response a signature over this nonce. The nonce MUST be generated using a random number generator compliant with functionality class NTG.1, PTG.3, or DRG.4 according to [24].

If other components are involved in the authorization of the contract key based on their signature, the respective private keys, used in the generation of these signatures, SHOULD be securely stored in HSMs. The usage of these keys MAY again be bound to the respective, local software states.

An ISO 15118 charging session is authorized with a signature by the contract key. Hence, authorization for this session is also bound to the authorization of the contract key. Because the contract key is underneath the storage key in the hierarchy, it is required to first provide authorization for the storage key in order to use the contract key. Hence, by binding the storage key to the local software state, the contract key is indirectly bound to the same restrictions and with that also the authorization for the charging process. As a result, the direct and indirect

method both ensure that the integrity EVCC's software state is always validated before a charging session is authorized.

As the OEM provisioning key can be used to request new contract keys, its usage MUST also be bound to the same authorization mechanisms as the contract key. This MAY again be realized directly (for the actual authorization of the provisioning key) or indirectly (for the authorization of any key above the provisioning key in the hierarchy). The binding to other EV components via a validation of their signatures MAY be omitted for the provisioning key.

3.3.4 Data at Rest Encryption

If the EVCC stores sensitive and/or privacy relevant logging information, its local storage MUST be encrypted and access to this data shall only be allowed to authenticated and authorized entities. The used keys MUST be protected by the EVCC's HSM. Authorization for the usage of the keys SHOULD be bound to the integrity of the local software state. Hence, if the system is compromised, the keys used to seal the privacy relevant information is no longer accessible. The authorization of the keys MAY be extended to protect against downgrade attacks, using the same method described in Section 3.3.3.

In the case where the EVCC saves security relevant logs that are not privacy relevant, these specific files SHOULD be only authenticity protected, using digital signatures, and not encrypted such that they are still accessible for forensic analysis in the case of a security incident. If this mechanism is used, the respective public key MUST be known to whoever is responsible for evaluating these logs so that they can validate their authenticity and the private key MUST be stored on the EVCC's HSM. Authorization for this key SHOULD be bound to the integrity of the local software state and MAY be extended to protect against downgrade attacks, using the same method described in Section 3.3.3.

Regarding the keys used for data at rest encryption/signatures, the general recommendations from [4] apply.

4 Requirements for the SECC

4.1 Overview

The basic use cases which the requirements in this section are meant to secure are listed in Section 4.2. They mainly consist of the installation of initial credentials on the SECC by its respective CPO, the SECC's role in ISO 15118 communication sessions, its backend communication and its local data storage. Section 4.3 lists requirements for the SECC regarding the security of these use cases. The requirements ensure a secure storage and usage of its credentials used for ISO 15118 and backend communication. Additionally, they provide a means for a CPO to validate the SECC's local software state via remote attestation over OCPP and a secure firmware update mechanism, while also ensuring confidentiality of its locally stored data.

4.2 Use Cases

The requirements for securing the SECC are centered around its role as server for ISO 15118 sessions and its communication with the CPO backend using OCPP. They are based on the following use cases:

- **CP Initialization:** Prior to its deployment, the CP's SECC is installed with the credentials required to authenticate itself towards an EVCC and the CPO backend.
- **ISO 15118 Sessions:** The SECC assumes the role of the server in ISO 15118 sessions and the underlying TLS connections. It authenticates itself towards the EVCC via the TLS handshake. During an ISO 15118 session, the SECC is responsible for session management, charging parameter negotiation, and validation of the authenticity of the EVCC requesting the charge. It also handles the necessary backend communication including checking if the authenticated EVCC is authorized to charge, transmitting necessary billing information, and forwarding certificate requests and response for contract credential installation.
- **OCPP Backend Communication:** The SECC uses OCPP [19] for communication with the backend infrastructure. OCPP supports usage of a secure TLS channel, where the backend server is authenticated during the TLS handshake. The SECC is authenticated either during the TLS handshake or

afterwards using HTTP basic. Backend communication is used for supporting the charging sessions (e.g., to check for charging authorization) and charge point management (e.g., remote firmware updates).

- **Local Authorization:** In certain situations the CP might authorize a charging process without communication with the backend (cf. [19] Section 3.4). For this, it maintains an authorization cache and/or a local authorization list. The authorization cache saves identifiers that were previously authorized by the backend. The local authorization list is sent by the backend and contains identifiers with their authorization status.
- **Logging of Events:** During its operation, the SECC might create and locally store different kinds of logging information. Some of these logs could potentially include sensitive or privacy relevant data (e.g., the authorization cache).

4.3 Securing the SECC

This section describes the requirements for an SECC regarding secure communication with the CPO backend over OCPP 1.6 [19, 20, 21] and secure storage of its OCPP and ISO 15118 credentials. Furthermore, requirements for the SECC for proving its software integrity based on remote attestation and for the secure storage of sensitive or privacy relevant data are provided. Independent of the used protocol, the communication between SECC and CPO backend MUST be secured (e.g., by using an IPSec VPN following the recommendations from [7]) and the SECC's private keys MUST be securely stored in an HSM meeting the requirements described in Section 5. For the rest of this section, the communication with the backend is assumed to use OCPP 1.6. Regarding the TLS connection underlying the OCPP 1.6 communication channel the general recommendations from [5] apply.

4.3.1 Deviations from the OCPP 1.6 Specification

While OCPP 1.6 defines some security mechanisms as optional, they are mandatory to meet the security goals of this technical guideline. These reinforced requirements are as follows:

- The use of TLS 1.2 or a higher version is REQUIRED.
- The use of TLS certificate based server authentication is REQUIRED.
- The key size of TLS RSA server certificates MUST be at least 2048 bits.
- The chosen cipher suit MUST be one of the recommended suites from [5].
- Cipher suites offering perfect forward secrecy MUST be preferred over ones that do not.

- Charge point authentication is REQUIRED.
- If charge point authentication is implemented using TLS client certificates, the general recommendations from [4] apply regarding the corresponding private key.
- Transport layer security for file transfer (firmware download or diagnostics upload) is REQUIRED (i.e., FTPS or HTTPS).

4.3.2 Key Storage

For protection against side-channel/run-time attacks, the private keys used to establish the ISO 15118 TLS connection MUST be securely stored on an HSM.

Client authentication in OCPP 1.6 uses either HTTP basic (OCPP1.6J [20]) or a TLS client certificate (OCPP1.6S [21]). The respective private values, i.e., 20 byte authorization key used for HTTP basic or the private key corresponding to the client certificate MUST be securely stored in an HSM which meets the requirements defined in Section 5.

4.3.3 Remote Attestation

To prove its software integrity to the CPO, the SECC MUST offer a remote attestation mechanism. The required messages SHOULD be implemented using the OCPP 1.6 DataTransfer request and response messages, i.e., the CPO sends a 128 bit nonce in the DataTransfer request and the SECC sends its authenticity protected measurement (e.g., signed hash of its software state and the nonce) in the DataTransfer response. The nonce MUST be generated using a random number generator compliant with functionality class NTG.1, PTG.3, or DRG.4 according to [24]. The CPO MUST securely store every possible, trusted software state of its respective charge points in order to verify if the received measurement indicates a trusted state.

The SECC's local software state MUST be measured in a measured boot. Optionally, measured boot MAY be replaced with a secure boot. The key used to provide authenticity protection of the measurements MUST be securely stored in the SECC's HSM and the key used to verify the authenticity MUST be announced to the CPO in an authentic manner and in case of a symmetric key also confidentiality protected. Regarding the key used to provide authenticity protection of the measurements, the general recommendations from [4] apply.

In case of measured boot, the CPO MUST perform a remote attestation of its charge points after every boot. To avoid the charge point sending more messages before its state has been verified, remote attestation SHOULD be performed before accepting the charge point's BootNotification. For this, the first BootNotification request from a charge point SHOULD be answered with the response *Pending* and an appropriate timeout for the next request. After this response,

the CPO requests the remote attestation using the DataTransfer message. Future BootNotifications should only be accepted after successful validation of the charge point's software state and rejected otherwise. The relevant steps for the described process of remote attestation after boot are shown in Figure 4.1.

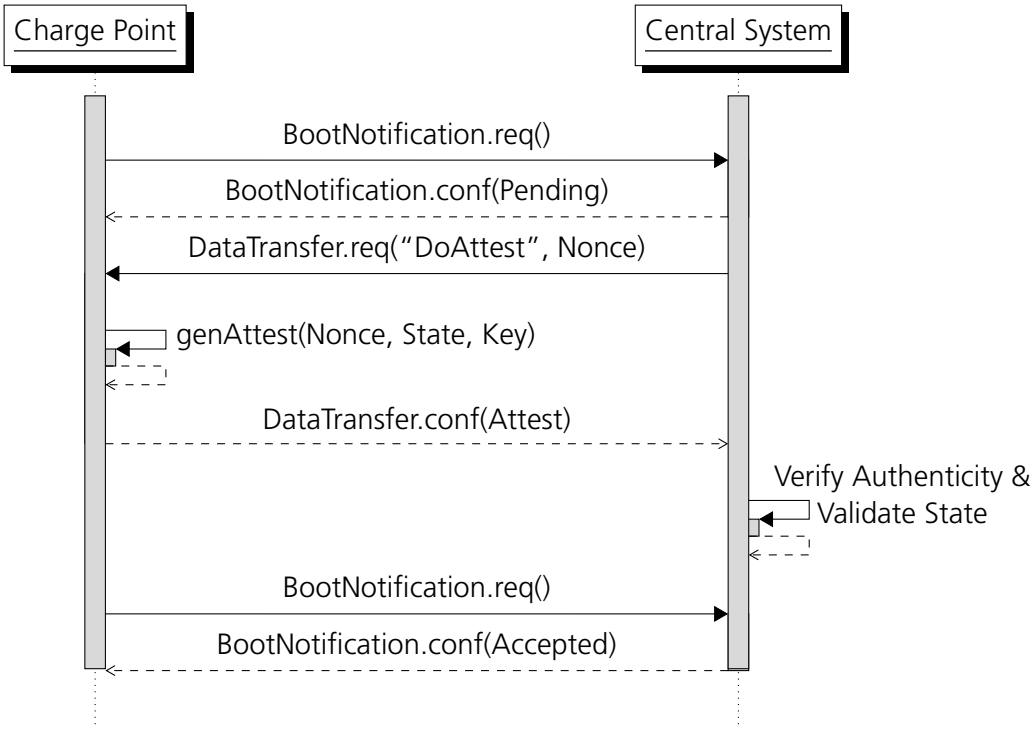

Figure 4.1: Remote Attestation over OCPP after Charge Point Reboot

However, since OCPP allows omitting the BootNotification on connection re-establishment [20, 21], the CPO might not know if the charge point has rebooted. For this reason, the CPO SHOULD perform a remote attestation of its charge points periodically. If the remote attestation of a charge point reveals an error, either because the software state is not trusted or the charge point failed to respond to multiple attestation requests in a row, the fault MUST be fixed without undue delay. If an error is reveled, the CPO MAY also treat the charge point's messages differently, e.g., by denying its boot notification, authorization and start transaction requests, until the next successful remote attestation.

4.3.4 Data at Rest Encryption

Since charge points save sensitive and privacy relevant data (e.g., the authorization cache or the local authorization list), their local storage MUST be encrypted and access to this data shall only be allowed to authenticated and authorized entities. The used keys MUST be protected by the EVCC's HSM. Authorization for the usage of the keys SHALL be bound to the integrity of the local software state. To enable secure firmware updates without the need to change the key,

this SHOULD be achieved similarly to the method described in Section 3.3.3, i.e., by binding it to any signed software state verifiable with a predefined public key. Also, the secret key SHOULD additionally be bound to a comparison of the current firmware version to a monotonically increasing counter (increased to the highest, valid firmware version) in order to provide downgrade protection.

In case the SECC stores security relevant logs, which are not privacy relevant, only integrity and authenticity MUST be ensured using digital signatures. The log data does not have to be encrypted in this case but must be accessible in the case of a security incident. If this mechanism is used, the respective public key MUST be known to whoever is responsible for evaluating these logs so that they can validate their integrity and authenticity. The private key MUST be stored in the SECC's HSM and access must be bound to the integrity of the local software state. For this, the methods allowing for secure firmware updates while providing downgrade protection are RECOMMENDED.

For the keys used for data at rest encryption/signatures, the general recommendations from [4] apply.

5 Requirements for the HSM used by EVCC and SECC

5.1 Overview

The HSM used by EVCC and SECC SHALL meet the security requirements defined in [8]. This document proposes a protection profile for the security module for the electric vehicle charging system and defines security objectives as well as security requirements, which shall be met by HSMs used by EVCCs and SECCs. Section 5.2 summarizes and specifies the relevant requirements for an HSM to comply with the use cases and requirements of Section 3, Section 4, and Section 6. Section 5.3 provides recommendations for implementing these requirements.

5.2 HSM Requirements

EVCC and SECC: The HSMs of both, EVCC and SECC are used for the secure, long-term storage of their credentials. For this, both HSMs MUST offer an Root of Trust for Storage (RTS) and MUST at least be certified under Common Criteria EAL 4. To enable all ISO 15118 functionality, both HSMs MUST offer support for EC keys on the secp256r1 curve (32 byte private keys) as well as the ECDH and ECDSA algorithms.

To provide integrity protection of the EVCC's and SECC's software, both HSMs MUST support a method for measuring the local software state during a measured boot, providing a Root of Trust for Measurement (RTM). Also, the authorization for usage of a key MUST be able to be bound to a trusted software state. To enable software updates without a need to change this key, the binding to software states SHOULD be possible via a public key used to validate the authenticity of any provided state which the locally measured state is then compared against. For downgrade protection, the HSMs MAY offer support for monotonically increasing counters and the possibility to bind authorizations to a comparison with the counter value.

EVCC: The EVCC's HSM SHALL support the ordering of its keys in a hierarchy such that a subordinate key can only be loaded and used by subsequently loading all keys above it in the hierarchy. Additionally, the HSM MUST be able to restrict the usage of its keys to authorized entities based on some kind of authorization mechanism (e.g., password based). Using a key MUST require authorization for

this key as well as all the respective authorizations for every previous key in the hierarchy.

For the provisioning of contract credentials, the EVCC's HSM is REQUIRED to offer a direct import functionality for encrypted contract keys, i.e., decryption is handled internally by the HSM during the import, such that the contract key never leaves the HSM in unencrypted form. Also, the EVCC'S HSM MUST support the ability to restrict the storage keys potential for general-purpose decryption so that it is not possible to decrypt contract keys without directly importing them.

In order to also bind the usage of the EVCC's contract keys to any other components involved in the charging process, the EVCC's HSM SHOULD support binding the authorization for key usage to the validation of an externally created signature.

SECC: For storage of the SECC's OCPP credentials, the HSM of a charge point supporting OCPP1.6J MUST provide secure storage of arbitrary 20 byte strings (either directly in the HSM or externally by sealing it with a key stored in the HSM). The SECC's HSM of a charge point supporting OCPP1.6S MAY need to offer support for additional algorithms from [4], in order to offer secure storage and usage of the SECC's private key used for client authentication during the TLS handshake.

To enable remote attestation, the SECC's HSM MUST, in addition to the RTM, provide a Root of Trust for Reporting (RTR) or work with an external RTR to prove the authenticity of its measurements to the CPO. The HSM MUST offer secure storage for the key used to provide authenticity protection of the measurements during remote attestation. For this, the SECC's HSM MAY need to offer support for additional algorithms from [4]. Optionally, the measured boot requirement of the SECC's HSM MAY be replaced with a secure boot.

5.3 Recommended HSM Implementation

To meet the requirements from Section 5.2 it is RECOMMENDED to use a hardware implementation of an HSM according to ISO 11889 [15, 12, 13, 14], the international standard for a TPM 2.0, or an HSM providing comparable functionality. ISO 11889 suits all mandatory and optional requirements for an HSM for both EVCC and SECC as follows:

- It offers RTM, RTS, and RTR (cf. [15] Section 9.4).
- Hardware TPM 2.0 implementations exist, which are certified according to Common Criteria with assurance level EAL 4 or higher (e.g., [2, 3] with EAL 4+).
- It supports the secp256r1 ECC curve (cf. [12] Section 7.4, called TPM_ECC_NIST_P256 which is equivalent to secp256r1 [18]).

- It meets the requirements for measured / secure boot (cf. [15] Section 9.5.5).
- It supports binding a key to the local software state (cf. [13] Section 24.6).
- It supports binding a key to a public key used to validate the authenticity of any provided state (cf. [13] Section 24.16).
- It offers monotonically increasing counters (cf. [13] Section 32.2).
- It supports binding a key to a comparison with a counter value (cf. [13] Section 24.9).
- It supports restricting key usage to authorized entities (cf. [15] Section 19.6.4).
- It supports ordering keys in a hierarchy such that loading a subordinate key requires subsequently loading and providing authorization for all keys above it in the hierarchy (cf. [15] Section 23).
- It offers a direct import functionality (cf. [13] Section 14.3).
- It supports restricting a key's potential for general-purpose decryption (cf. [15] Section 25.1.5).
- It supports binding the authorization for key usage to the validation of a signature (cf. [13] Section 24.3).
- It supports binding the authorization for key usage to multiple authorization mechanisms (cf. [15] Section 19.7).
- It supports sealing of arbitrary data (cf. [13] Section 13.1).
- It supports remote attestation (cf. [13] Section 19.4).
- Regarding the SECC's keys used for client authentication during the OCPP1.6S TLS handshake and for authentication of its remote attestations, ISO 11889 supports many of the algorithms, key lengths, and parameters recommended by [4] (cf. [12] Section 7.3).

6 Requirements for the ISO 15118 communication between EVCC and SECC

6.1 Overview

The basic use cases which the requirements in this section are meant to secure are listed in Section 6.2. They mainly consist of the establishment of a secure communication channel using the TLS protocol, the authentication of the SECC during the TLS handshake, and the authentication of the EVCC on application layer. Section 6.3 lists requirements for the EVCC and SECC regarding the security of these use cases. The requirements ensure a secure establishment of the communication session and secure storage and usage of the involved credentials.

6.2 Use Cases

The requirements for securing the communication interface between EVCC and SECC are centered around the ISO 15118 specification and based on the following use cases:

- **TLS Channel Establishment:** TLS is mandatory for the ISO 15118 PnC identification mode. The TLS handshake uses unilateral, server-side authentication. For this, the SECC's leaf certificate traces back to the V2G root certificate, preinstalled in the EVCC. After the TLS communication channel is established, all ISO 15118 communication is sent via the TLS tunnel, i.e., authenticity, integrity, and confidentiality are protected using the derived session keys.
- **EVCC Authentication:** The EVCC is authenticated during an ISO 15118 communication session, using an application layer challenge-response mechanism. The SECC sends a challenge to the EVCC. The EVCC uses its contract certificate private key to sign the provided challenge and sends the response back to the SECC. The SECC validates the signature using the public key of of the contract certificate (and checks the validity of the certificate with its chain if not done before). During the charging session, the EVCC signs metering receipts at the SECC's request, also using its contract certificate private key.

6.3 Securing the EVCC-SECC Interface

Regarding the TLS connection, underlying the ISO 15118 communication channel, the general recommendations from [5] apply. For protection against side-channel/run-time attacks, the SECC's private key used to establish the ISO 15118 TLS 1.2 connection MUST be securely stored in its HSM. ISO 15118 uses a unilateral, server-side TLS authentication with the cipher suites (cf. [11] Section 7.7.3.4):

1. TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
2. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256.

While SECC's MUST support both versions to accommodate for legacy systems, EVCC's SHOULD only support the second variant, with ECDHE for key agreement, since it is the only one providing perfect forward secrecy (cf. [5] Section 3.3.1.1). In case where EVCC and SECC each support both versions, the second one MUST always be preferred. Due to their short lifespan, the ephemeral EC and AES session keys MAY be used outside of the HSM. However, ephemeral and session keys MUST be permanently deleted as soon as they are no longer needed.

Since ISO 15118 uses no client side TLS authentication, none of the EVCC's static private keys are involved in the TLS connection. Hence, only short lived session keys are used and there are no additional requirement towards the EVCC's HSM with regard to the TLS connection. The EVCC's credentials used during the ISO 15118 session (e.g., for PnC authentication or authentication of certificate requests) MUST be securely stored in its HSM.

The current draft for the second edition of ISO 15118, ISO 15118-20, changes the used block cipher mode of operation in the TLS cipher suites from CBC to GCM [16]. Hence, the new cipher suites are (cf. [16] Section 7.7.3.4):

1. TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
2. TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256.

Since the second cipher suite is fully compatible with TLS 1.3 [23] and ISO 15118-20 explicitly allows the optional support of TLS versions higher than 1.2 (cf. [16] Requirement V2G2-ED2-1521), changing from TLS 1.2 to TLS 1.3 is easily possible by using the following TLS 1.3 cipher suite:

- TLS_AES_128_GCM_SHA256

and setting the following values in the extensions:

- ecdsa_secp256r1_sha256 (signature_algorithms extension)
- secp256r1 (for ECDHE in supported_groups extension).

While EVCC and SECC MUST still offer support for TLS version 1.2 in order to meet the ISO 15118 requirements, it is RECOMMENDED that they additionally support TLS 1.3. It is also RECOMMENDED to use TLS 1.3 instead of TLS 1.2 for the ISO 15118 communication whenever possible. If TLS 1.3 is used, the general recommendations from [5] apply and the SECC's private key used to establish the TLS 1.3 connection MUST be securely stored in its HSM.

Bibliography

[1] Bradner, Scott O.: *Key words for use in RFCs to Indicate Requirement Levels*. RFC 2119, March 1997. https://rfc-editor.org/rfc/rfc2119.txt. 1

[2] BSI: *BSI-DSZ-CC-1021-V2-2017*. Certification Report for Infineon Technologies AG Trusted PlatformModule SLB9670_2.0, v7.62.3126.00, v7.62.3127.00, Federal Office for Information Security, August 2017. 21

[3] BSI: *BSI-DSZ-CC-1100-2018*. Certification Report for Infineon Technologies AG OPTIGA™ Trusted Platform Module SLI9670_2.0 and SLM9670_-2.0,v13.11.4555.00, Federal Office for Information Security, December 2018. 21

[4] BSI: *Cryptographic Mechanisms: Recommendations and Key Lengths: Part 1*. Technical Guideline TR-02102-1, Federal Office for Information Security, February 2019. 14, 17, 19, 21, 22

[5] BSI: *Cryptographic Mechanisms: Recommendations and Key Lengths: Part 2 – Use of Transport Layer Security (TLS)*. Technical Guideline TR-02102-2, Federal Office for Information Security, February 2019. 16, 24, 25

[6] Deutsche Kommission Elektrotechnik Elektronik Informationstechnik (DKE): *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 edition, October 2017. https://www.vde-verlag.de/standards/0800432/vde-ar-e-2802-100-1-anwendungsregel-2017-10.html. 4, 5

[7] Frankel, Sheila E., Karen Kent, Ryan Lewkowski, Angela Orebaugh, Ronald Ritchey, and Steven Sharma: *SP 800-77. Guide to IPsec VPNs: Recommendations of the National Institute of Standards and Technology*. Technical report, Gaithersburg, MD, United States, December 2005. 16

[8] Fuchs, Andreas, Christoph Krauß, Norman Lahr, and Richard Petri: *Security Module for the Electric Vehicle Charging System - Proposal for a Protection Profile*. Version 1.0, Fraunhofer Institute for Secure Information Technology (SIT), November 2019. 20

[9] Hubject GmbH: *Open InterCharge Protocol for Emobility Service Provider*, version 2.1 edition, May 2016. https://www.hubject.com. 3, 5

[10] ISO/IEC: *Road vehicles – vehicle to grid communication interface – part 1: General information and use-case definition*. ISO Standard 15118-1:2013, International Organization for Standardization, Geneva, Switzerland, April 2013. 1, 2, 3, 4, 5

[11] ISO/IEC: *Road vehicles – vehicle-to-grid communication interface – part 2: Network and application protocol requirements*. ISO Standard 15118-2:2014, International Organization for Standardization, Geneva, Switzerland, April 2014. 2, 3, 4, 5, 12, 24

[12] ISO/IEC: *Information technology – trusted platform module library – part 2: Structures*. ISO Standard 11889-2:2015, International Organization for Standardization, Geneva, Switzerland, December 2015. 21, 22

[13] ISO/IEC: *Information technology – trusted platform module library – part 3: Commands*. ISO Standard 11889-3:2015, International Organization for Standardization, Geneva, Switzerland, December 2015. 21, 22

[14] ISO/IEC: *Information technology – trusted platform module library – part 4: Supporting routines*. ISO Standard 11889-4:2015, International Organization for Standardization, Geneva, Switzerland, December 2015. 21

[15] ISO/IEC: *Information technology – trusted platform module library – part 1: Architecture*. ISO Standard 11889-1:2015, International Organization for Standardization, Geneva, Switzerland, April 2016. 21, 22

[16] ISO/IEC: *Road vehicles – vehicle to grid communication interface – part 2: Network and application protocol requirements*. ISO Standard - DRAFT 15118-2:2018, International Organization for Standardization, Geneva, Switzerland, December 2018. 24

[17] National Institute of Standards and Technology: *Framework for Improving Critical Infrastructure Cybersecurity*. Version 1.1, April 2018. 6

[18] Nir, Yoav, Simon Josefsson, and Manuel Pégourié-Gonnard: *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier*. RFC 8422, August 2018. `https://rfc-editor.org/rfc/rfc8422.txt`. 21

[19] Open Charge Alliance: *Open Charge Point Protocol 1.6*, version 1.6 edition, 2015. `http://www.openchargealliance.org/protocols/ocpp/ocpp-16/`. 3, 4, 5, 15, 16

[20] Open Charge Alliance: *Open Charge Point Protocol JSON 1.6*, version 1.6 edition, 2015. `http://www.openchargealliance.org/protocols/ocpp/ocpp-16/`. 3, 16, 17, 18

[21] Open Charge Alliance: *Open Charge Point Protocol SOAP 1.6*, version 1.6 edition, 2015. `http://www.openchargealliance.org/protocols/ocpp/ocpp-16/`. 3, 16, 17, 18

[22] PCI Security Standards Council: *Payment Card Industry (PCI) Data Security Standard - Requirements and Security Assessment Procedures*. Version 3.2.1, May 2018. 6

[23] Rescorla, Eric: *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446, August 2018. `https://rfc-editor.org/rfc/rfc8446.txt`. 24

[24] Wolfgang Killmann, Werner Schindler: *A proposal for: Functionality classes for random number generators*. Version 2.0, Federal Office for Information Security (BSI), September 2011. 13, 17