European Network for Cyber Security

Commissioned by ElaadNL

EV Charging Systems

Security Requirements

Version 1.01 - August 2017

Contents

1 Introduction

This catalog describes security requirements for Electric Vehicle charging systems. Two sets of requirements are included:

1. A set of requirement for the *procurement of Charge Point*. This set includes requirements to make sure the Charge Point itself is secure (Section 2), that it has all functionality needed to set up secure operational processes (Section 3), that its Vendor takes measures to ensure its security throughout its lifecycle (Section 4), and that measures are taken to assure that security measures have been implemented well (Section 5).
2. A set of requirements *for secure communications between the Charge Point Operator (CPO) and Distribution System Operator (DSO)*. These requirements can be used as part of the security requirements when new server systems are procured or set up.

The definition of the requirements is based on the results of the Threat Assessment [2], which identified the threats and possible attacks related to EV charging systems. Each requirement is justified by one or more possible threats identified.

These requirements have been developed by the European Network for Cyber Security (ENCS) for ElaadNL. ElaadNL intends to use and promote the requirements as the basis for future development.

Questions regarding these requirements can be sent to: Harm.van.den.Brink@elaad.nl

1.1 Scope

The security requirements for the procurement of Charge Point cover the externally reachable interfaces (see the architecture document [1]), that is:

1. the WAN interface,
2. the Maintenance interface, and
3. the User Authentication (UA) interface.

These interfaces are located on the Local Controller and Authentication terminal. For each requirement it is indicated for which of these two devices it applies. In addition, the requirements cover secure firmware updates for the EVSE.

The communication between the CPO and DSO concerns the CPO interface.

The requirements are device-specific and are set to be fulfilled by the vendor. The requirements do not address operational security, for example operational requirements for the Charge Point Operator or Distribution System Operators are not included. However, the technical functionality needed for secure operations is included in the requirements, and for selected requirements operational recommendations are given.

1.1.1 Interoperability Requirements

The requirements are formulated in a technology and protocol independent manner. In this way they can be applied to many different types of the devices included in the EV Charging System. Charge Point Operators will choose different communication technologies and protocols, and different software systems based on their particular situation. The requirements in this document provide security for all possible choices.

Users of the requirements may want to complement these security requirements with interoperability requirements. Some specific technologies may be required for integration into a larger infrastructure.

Examples where interoperability requirements may be needed are:

- The communication security requirements in section 2.3. To implement these requirements, the devices need to use the same protocol as the software connecting to it. Often protocols such as IPsec or TLS are used. Interoperability requirements may be needed on the version and configuration of these protocols.
- The logging requirements in section 3.2. More and more logging and event information is being imported in Security Information and Event Management (SIEM) systems. The SIEM may put particular requirements on the protocol used to send the data and the format of the data.
- The requirements for time synchronization in the logging requirements in section 3.2. Different technologies are available to provide this feature, such as NTP and GPS. If a Charge Point Operator is already using one of these technologies, they may require to use the same technology.

1.2 Architecture

Electric Vehicle charging systems are used to supply energy for charging Electric Vehicle. The EV charging system groups multiple functions.

The EV Charging system is composed of Charge Points, Charge Point Operators and Distribution System operators (DSOs).

The Charge Point plays the charging role in EV charging system by supplying energy from the DSO to the Electric Vehicle.

The Charge Point has multiple other functions such as:

- Providing and controlling the energy to the EV using the Electric Vehicle Supply Equipment (EVSE) component
- Collecting the measurements from the meter for each charge of an Electric Vehicle.
- Identifying and authorizing EV users via user authentication component
- Enabling remote capabilities (e.g. adjustment of the maximum energy allowed by the Charge Point) to the Charge Point via the Local Controller component over WAN.

The Charge Point Operator's (CPO) role is:

- to give permission to the EV user to charge
- to gather data, processes, and measurements

- to send energy limits to control the energy flow allowed between the Charge Point and the EV from the data given by the DSO.
- To supply the connection to the CPO
- To connect the CPO
- To ensure power supply stability
- To forecast… (not really done right now)

The DSO's role is:

- To forecast the available capacity
- To ensure power supply stability

Figure 1 illustrates the architecture of the EV Charging Systems that are in scope of this project.

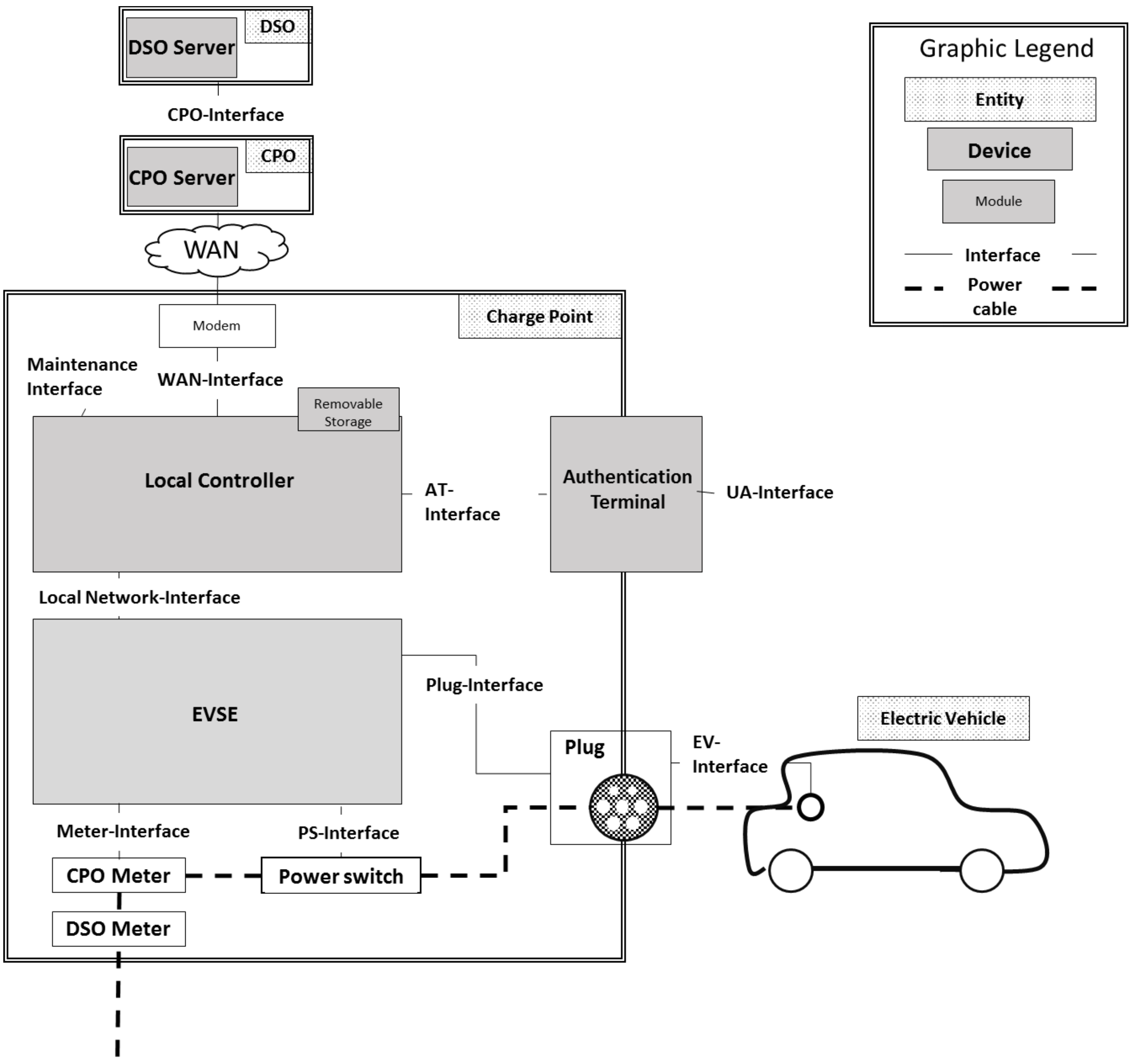

Figure 1: EV Charging System Architecture.

The security requirements document [3] covers three areas (given by the grey boxes):

1. *The externally reachable interfaces on the Charge Point*: the WAN interface, the Maintenance interface, and the User Authentication (UA) interface. These interfaces are located on the Local Controller and Authentication terminal.
2. *The firmware updates on the EVSE*. Requirements are included for the verification of firmware signatures to allow secure firmware updates on the EVSE.
3. *The communication between the CPO and DSO.* This concerns the CPO and DSO server systems. It is to be noted that sustainable energy producers that want to control the energy load on the Electric Vehicle environment will have to comply to the same requirements applicable to the DSO system.

All other components, shown in the white boxes, are out of the scope of these security requirements.

Note in particular that the internal interfaces in the Charge Point are not covered by the security requirements. This reflects the current situation in which most of these interfaces use serial protocols with no security features. This exclusion of these interfaces implies that the inside of the Charge Point is a trusted environment: anyone with physical access to the internal systems can compromise the Charge Point. Physical security measures are included in the requirements to prevent and detect unauthorized access to the Charge Point internals.

The EV Charging System Architecture reference various items in the Graphic Legend:

- An **Entity** represents a main part of the EV charging system.
- A **Device** identifies the component included in the EV charging system. A device is can contain Modules and can have Interfaces to communicate with other devices.
- A **Module** identifies the physical part of the Device where important functionalities are to be found.
- An **Interface** defines the communication link between two Devices. The list of the interfaces and the types of communication are defined below.

1.3 How to Read the Requirements

Each requirement is labelled with an identifier (Req.ID) and consists of the following five items:

- **Device:** The *Device* category defines a list of devices for which the Minimum Requirement, Awarding Criteria and Recommended Assurance applies.
- **Minimum Requirement:** A *mandatory requirement* is a compulsory function that an entity, device, or module must perform. Statements in the requirements of this document are compulsory for the vendor.
- **Awarding Criteria:** A*warding Criteria* are weighted and scored in the evaluation. The weights and scores are not defined in this document but will be set by the requesting organization.
- **Recommended Assurance:** *Recommended assurance* provides guidance for quality control. The vendor can see how the implementation of the requirement will

be tested in a standard testing facility. Appendix A provides brief remarks and references concerning the most common testing procedures.

Items may be left out for a particular requirement if they are not used.

The categories Minimum Requirement, Awarding Criteria, Recommended Assurance may refer to *Device*. For each, *Device* listed in the category Device, the Minimum Requirement, Awarding Criteria, and Recommended Assurance applies.

The CPO and DSO requirements in Section 6 are labelled with an identifier and an interface. The following name are used as identifers:

- **Req.ID.CPO** which stands for the CPO Server identified in the Architecture chapter of this document or in the Architecture document [1]
- **Req.ID.DSO** which stands for the DSO Server identified in the Architecture chapter of this document or in the Architecture document [1]

After these five items, further clarification on the requirement is given. The clarification can define certain terms, give examples of what is and is not allowed by the requirements, or give a recommendation on implementing the requirement. A requirement does not have to be implemented as in the recommendation, as long as the Vendor provides a good justification on why their implementation meets the requirement (see requirement SUR.01 in Section 5).

The requirements use standard terminology from security and EV charging where possible. If there is a possibility for confusion about a term, it will be defined in the clarification of the first requirement where it is used, and printed in bold there. A glossary of terms is also provided at the end of the document.

1.3.1 Wording

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 RFC 2119 [3]:

- *MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.*
- *MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.*
- *SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.*
- *SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.*

1.4 Requirements Categories

The requirements are divided in the following categories:

- **Future Proof Design (SFR):** This category of requirements uses the SFR identifier. These requirements aim at preventing lack of capabilities for future security updates.
- **Cryptographic Algorithms and Protocols (SPR):** This category of requirements uses the SPR identifier. There requirements aim at describing the cryptographic algorithms, key lengths, pseudo random generator allowed to use.
- **Communication Security (SCR):** This category of requirements uses the SCR identifier. These requirements aim at defining the necessary security mechanisms to implement for an end-to-end security for the EV Charging System.
- **System Hardening (SHR):** This category of requirements uses the SHR identifier. These requirements aim at providing hardening mechanisms for the Device.
- **Resilience (SRR):** This category of requirements uses the SRR identifier. These requirements aim at preventing issues due to misuse of the Device or the Interface.
- **Access Control (SAR):** This category of requirements uses the SAR identifier. These requirements aim at properly defining the Authorization mechanism on the Device or on its Interface.
- **Logging (SLR):** This category of requirements uses the SLR identifier. These requirements aim at defining the detection mechanisms to put in place in order to identify security issues that might occur on the Device or Interface.
- **Product Lifecycle and Governance (SDR):** This category of requirements uses the SDR identifier. These requirements aim at defining the processes used for developing, manufacturing, and provisioning of the EV charging system *Devices* in a secure way.
- **Assurance (SUR):** This category of requirements uses the SUR identifier. These requirements aim at describing the measures the Vendor should take to make sure the EV charging system *Devices* will work securely.

1.5 Stakeholders

The stakeholders concerned with the procurement and product lifecycle of the EV charging systems are Purchasers and Vendors.

This document uses the term Purchaser as replacement for charge point operator, distribution system operator (DSO), utility, grid operator or similar. The term Vendor stands for the party that sells the EV charging systems. The document does not distinguish between a vendor and a manufacturer in case these are two separate entities. Ultimately, the Vendor is held responsible for the security features of the product, i.e., the local controller. In particular, the Vendor has to ensure that all components procured from a supplier satisfy the requirements in this document.

2 EV Charging Security Requirements

This section contains the technical requirements to keep the EV charging system secure. Care has been taken to align these requirements with common standards and best practices for security for devices used in the industrial control systems domain, such as the IEC 60068-2-75 [4], the IEC 62351 series [11], and the IEC 62443 series (former ISA-99) [10]. Moreover, the recommendations used for cryptographic algorithms are based on recommendations documents, such as ENISA document on algorithms, parameters, and key sizes [12] or BSI document [37].

2.1 Future-Proof Design

The requirements in this section concern future-proof designs for the EV charging system *Devices*. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SFR.**".

SFR.01 Future-Proof Design

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device SHALL have sufficient reserves in memory and computing power to allow updates to security functions that security experts anticipate are necessary during the Device's lifecycle.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Testing the performance of the Device for algorithms and protocols anticipated for future use.

In this document a **security function** refers to any function on the *Device* that is needed for it to be operated securely. Security functions include access control, authentication, and encryption. All functions needed to implement the security requirements in this document shall be considered as security functions.

There are several sources of expert forecasts on what security functions are needed in the future. It is recommended that Vendors consult these sources when determining which algorithms and protocols are needed in the future.

The ENISA documents on algorithms, parameters, and key sizes [12] marks some algorithms as suitable for future use, while others are only suitable for legacy use. If legacy algorithms are used by the Device, there should be sufficient resources to update it to an algorithm in the same category suitable for future use.

Recommendations on which key sizes provide sufficient security in the future are available from e.g. NIST [18], BIS [38], and ANSSI [38]. One way to show that sufficient computational resources are available, is to show that the Device can support the key sizes required by these document at the end of the Device's lifecycle.

The German Federal Office for Information Security (BSI) classifies IPsec and IKEv2 options in [15]; for each option BSI states a year until which the option is considered secure. The label "2021+" means that the option is considered secure until the year 2021 and beyond.

SFR.02 Hardware Design

Minimum Requirement	1. The RFID reader of the Device SHALL be easily and fully replaceable in case new standards require changes of this part.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Testing the performance of the Device for algorithms and protocols anticipated for future use.

SFR.03 Remote Firmware Updates

Device	• Local Controller
Minimum Requirements	1. The Device SHALL support updating all security functions through remote firmware updates.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor.

This requirement does not forbid updates of security functions over the Maintenance interface. Such a requirement would be operational and is left to the Charge Point Operator to decide. SCR.03 details verification of the integrity of firmware updates.

2.2 Cryptographic Algorithms and Protocols

The requirements in this section concern how to choose cryptographic tools and key lengths. Requirements are grouped into different items. Each item has a unique identifier with prefix **"SPR."**.

SPR.01 Cryptographic Algorithms and Key Lengths

Device	• Local Controller • EVSE • Authentication Terminal

Minimum Requirements	1. For security functions, the Device SHALL use only cryptographic algorithms for which a description is publicly available, and which have been thoroughly reviewed by independent cryptographers. 2. For security functions the Device SHALL not use cryptographic algorithms, protocols, and parameters if there are vulnerabilities known for them. 3. If for a security function algorithms are available in [12], the Device SHALL use one of these algorithms. 4. The Device SHALL use from [12] only those cryptographic algorithms, and parameters considered suitable for legacy or future use. 5. The Device SHALL use the algorithms in [12] implemented exactly as they are described there without any modifications.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor can be used to establish that only allowed cryptographic algorithms, protocols, and parameters are used. • Functional security tests can be used to verify that the algorithms are implemented as described in [12]. • Cryptographic primitives can be certified with the NIST Cryptographic Algorithm Validation Program (CAVP) [23].

A cryptographic protocol is a protocol used for security functions, such as authentication protecting confidentiality or integrity. Cryptographic protocols are implemented using cryptographic algorithms, such as symmetric and asymmetric ciphers, and hash functions. The cryptographic algorithms again depend on certain cryptographic parameters. The most well-known example is the key size. If the key size for an algorithm is too small an algorithm becomes vulnerable to brute-force attacks. Correct choices for other cryptographic parameters, such as the initialization vector, are equally important for the secure functioning of a protocol.

Vulnerabilities are considered known if they are in a public vulnerability database, or if an advisory on them has been published. The ENISA report [12] provides a good overview of the state-of-the-art for cryptographic primitives such as block ciphers, cryptographic hash functions, stream ciphers, public-key primitives, and a key size analysis. The report is updated annually to be in accordance with technical and scientific progress. When an algorithm is marked as suitable for legacy use in this reports, it means that there are no known vulnerabilities and the algorithm is considered good for current use. When it is marked at suitable for future use, it is expected to remain secure for 10 to 50 years.

Some algorithms in [12] are not even allowed for legacy use, and are marked with an "X" in the legacy column. Such algorithms are broken and considered insecure. They must not be used on the *Device* for security functions.

Examples are:

- The MD5 hash algorithm: an attacker can construct two distinct files with the same MD5 hash value. In particular, it would be possible to produce a second firmware image with different content but matching hash value.
- The RC4 stream cipher: encryption can be broken due to biases in the key stream.
- It is allowed to use algorithms for which vulnerabilities are known if they are not used for security functions. For instance, cyclic redundancy checks (CRC) codes can be used by the *Device* to detect accidental errors in the transmission of a message. They should however not be used to check against deliberate modifications by attackers (as required in SIR.02) as there are vulnerabilities known for them.

To interpret the requirement, it is important to distinguish between cryptographic protocols and communication protocols, such as TLS, IPsec. Communication protocols usually use several cryptographic protocols to implement their security features. Often they offer different options for each feature. For instance, the TLS protocol allows both RSA and (elliptic curve) Diffie-Hellman for key exchange, and allows for different key sizes for each protocol. If vulnerabilities are known for some of the cryptographic options allowed by a communication protocol, it does not mean the communication protocol should not be used. Instead, only secure options should be used, and others disabled.

For several communication protocols commonly used, there are vulnerabilities known for all the cryptographic protocols used in older protocol versions. In that case the older protocol version should not be used. Examples are:

- All versions of SSL and TLS versions before 1.2 have known vulnerabilities. If the *Device* uses TLS, it must use version 1.2 or greater.
- SNMP versions before version 3 have known vulnerabilities.

Communication protocols with known vulnerabilities can be used if they are encapsulated in other protocols that provide the security functions. The most common case is that vulnerable protocols are encapsulated in secure network or transport layer protocols, such as IPsec or TLS.

Many industrial protocols, such as Modbus, do not implement any security. Such protocols should therefore always be encapsulated in secure lower layer protocols.

SPR.02 Cryptographic Random Number Generation

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device SHALL use a dedicated cryptographic pseudo-random number generator, as defined in FIPS 186-2 [24], FIPS 140-2 (Annex C) [26], AIS 20 [26], or AIS 31 [27], to generate random numbers used for security functions. The Device SHALL use the algorithms in [12] implemented exactly as they are described there without any modifications.

Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Proof of the implementation could be the reports of a standardized test procedure such as the NIST Cryptographic Algorithm Validation Program (CAVP) [23]. • NIST SP 800-22 [35] provides a standardized test suite to look for biases found in non-cryptographic random number generator during a black-box test.

Random values are used for security function for instance in the generation of digital signatures and cryptographic keys, or in authentication protocols.

The basic random number generators in many programming languages, such as the rand() function in the C programming language, do not satisfy the requirements in the mentioned standards. For Linux-based systems one can instead use /dev/random. The German BSI recommends in [28] to use kernel versions starting from 2.6.21.5, 3.2, 3.5, 3.6 and 3.7. It is recommended to monitor vulnerabilities in implementations and update kernels accordingly.

ENISA provides further requirements on pseudo-randomness generation in [12].

SPR.03 Key Management

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device MUST support remote updates of all credentials and cryptographic keys. 2. The Device MUST support limiting the duration of a session to a time length that is configurable by the purchaser.
Awarding Criteria	3. The Device SHOULD support establishing a fresh key for each communication session. 4. The Device SHOULD support using different keys for different services and applications.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Functional tests can be used to establish the functionality is present on the Device.

Establishing a session key can only be done if the *Device* and the hosts it communicates with use the same protocol. Hence, there may be interoperability requirements. Because the *Device* supports key updates, it is possible to give each similar *Device* individual keys. It is strongly recommended that this is done by Purchasers operating the *Device.*

Pre-shared keys are considered less secure than session keys. Attacks on cryptographic algorithms often require a large amount of encrypted data. By using a session key, the amount of data encrypted with one key is limited. Therefore, it is preferred that a fresh key is generated for each session. In this context, a key should be considered fresh if it was generated by a cryptographic random number generator (as defined in SPR.02) or a cryptographic key exchange algorithm (such as Diffie-Hellman key exchange), and was not used before.

Using TLS has the advantage that it allows different keys for different services and applications, so that awarding criterion 4 is met.

SPR.04 Cryptographic Versioning

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device MUST implement version identifiers for the communication protocol used which implement the security functionalities. 2. The Device MUST be able to configure the minimum version of the protocol that is allowed, and reject connections with older protocol versions.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Functional tests can be used to establish the functionality is present on the Device.

Cryptographic versioning of protocols is crucial to allow for updates of security in the field. For instance, while a protocol is being updated with firmware updates, the CPO Server will communicate with Charge Points with different protocol versions. The CPO Server needs to be able to know which Charge Point uses which version, and which protocol version. Otherwise this type of update is not possible.

Protocols such as TLS, IPSec, and SSH already implement this type of versioning.

2.3 Communication Security

The requirements in this section concern communication security for the EV charging system *Devices*. Each item has a unique identifier with prefix “**SCR**.”.

The communication security only concerns the communication from devices in the Charge Point with external systems. The communication interfaces within the Charge Points, such as that between the Local Controller and the Authentication Terminal, typically rely on USB, UART or serial connectivity. These communication channel are not required to be protected by cryptographic measures. But requirement SHR.05 below does ask that the physical perimeter of the Charge Point is protected.

SCR.01 Confidentiality

Device	• Local Controller
Minimum Requirements	1. The Device SHALL protect the confidentiality of communication on the WAN interface by encrypting it using a protocol allowed by SPR.01. 2. The Device SHALL store passwords together with a salt using a cryptographic hash function allowed by SPR.01.
Recommended Assurance	• This requirement is verified in a functional security test. The test should in particular ensure that the allowed cryptographic algorithms are supported and that disallowed algorithms are rejected.

The default option for encryption in the OCPP protocol, used on the WAN interface, is to use TLS. Using other solutions, such as VPNs, is also allowed, as long as they meet requirement SCR.01.

Encryption on the Maintenance and Local Network interfaces is not required, as intercepting traffic on them is not possible without local access in the Charge Point, and the value of the information that can be captured in one Charge Point is low.

Special protection is required for passwords, because it should be the only truly confidential information stored on the *Role* or *Device.* The requirements in this document are set up to allow for different keys for each *Device.* If the Purchaser indeed uses different keys in operations, attackers will benefit little from getting the keys out of the *Device.* They must already compromise the *Device* to get the key, and they cannot use the keys on other *Devices.*

It is still recommended to use different passwords for each *Device.* Attackers that compromise the *Device* may still acquire passwords by capturing them when they are sent to the *Device.* Using different passwords does require support from the tools used for maintenance, and the central servers to remember the passwords. Engineers and operators cannot be expected to remember passwords for hundreds of *Devices.*

The requirement does not apply to the Authentication Terminal. The architecture assumes that the Authentication Terminal only communicates with external systems indirectly through the Local Controller. The Local Controller is responsible for the security of these communications.

SCR.02 Message Integrity

Device	• Local Controller
Minimum Requirements	1. The Device SHALL verify the integrity of application layer messages received on the WAN and Maintenance interface using a message authentication algorithm allowed by SPR.01.

	2. If the Device detects that a message has been modified or if it cannot verify the integrity of the message, it SHALL reject or drop the message. 3. The Device SHALL allow parties it communicates with on the WAN interface or Maintenance to verify the integrity of application layer messages it sends by using a message authentication algorithm allowed by SPR.01.
Awarding Criteria	4. The Device SHOULD verify the cryptographic integrity of messages received on the Local Network interface. 5. The Device SHOULD allow parties it communicates with on the Local Network interface to verify the integrity of application layer messages it sends by using a message authentication algorithm allowed by SPR.01.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Functional tests can be used to verify that the Device supports the required functionality. • Carrying out a penetration test can be used to determine if the Device verifies message integrity under all conditions.

Message integrity is usually verified using a message authentication code (MAC) or a block cipher in authenticated encryption mode, such as Galois Counter Mode (GCM). Algorithms for these are available in [12]. To be able to verify the integrity of an application layer message, the entire message should be given as input to the message authentication algorithm. No message fields should be left out.

The integrity of messages without application layer payload, such as acknowledgements, does not have to be protected. Headers from lower layer protocols also do not have to be protected. If these headers however include counters or information on the message's source, this information may still require integrity protection to meet requirements SCR.04 and SCR.05.

If IPsec is used to fulfil this requirement the Encapsulating Security Payload (ESP) should use one of the authenticated cipher modes (AES-GCM or AES-CCM). Alternatively, the Authentication Header (AH) should be configured using one of the allowed cryptographic algorithms (see SPR.01).

This requirement concerns cryptographic message integrity. CRC checksums do not fulfil the requirement. They are not allowed by requirement SPR.01.

A message is dropped if the *Device* does not send a reply. A message is rejected if the *Device* replies with an error message or NACK.

On the Local Network interface message integrity checks are not a minimum requirement. To exploit the lack of integrity checks, attackers also first need to have access to the networks in the Charge Point.

This requirement does not apply to the Authentication Terminal, for the same reason as for requirement SCR.01.

SCR.03 Firmware Integrity

Device	• Local Controller • EVSE • Authentication Terminal
Minimum Requirements	1. The Device SHALL verify the integrity of firmware images before they are applied. 2. The Device SHALL reject firmware updates if it detects the firmware has been modified, or it cannot verify the firmware's integrity.
Recommended Assurance	• The functional requirement can be verified by testing the implemented firmware-update functions.

Firmware integrity is usually verified by calculating a hash value of the firmware. Hash functions are described in the ENISA document [12].

SCR.04 Message Freshness

Device	• Local Controller
Minimum Requirements	1. The Device SHALL be able to detect replay attacks on the WAN interface. 2. If the Device detects that a message is replayed, it MUST reject or drop the message.
Awarding Criteria	3. The Device SHOULD be able to detect replay attacks all the interfaces.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used to protect against replay attacks. • Functional testing can be used to verify if the mechanisms are indeed implemented.

To prevent replay attacks all messages should be secured by one of the following means:

- By adding a counter.
- By adding an authenticated nonce. It is essential that the nonce is authenticated using a MAC algorithm.

VPN technologies such as IPsec need to explicitly enable replay protection in combination with message authentication (SCR.02).

This requirement does not apply to the Authentication Terminal, for the same reason as for requirement SCR.01.

SCR.05 Message Authentication

Device	• Local Controller
Minimum Requirements	1. The Device SHALL be able to determine that the source of a message is a specific host in the EV Charging system.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used for message authentication. • Functional testing can be used to verify if the mechanisms are indeed implemented. • Penetration tests can be used to ascertain that attackers cannot bypass the authentication mechanisms.

Authentication concerns being able to determine the source of a message. There are different levels of detail possible here. The source can be a host in the network, such as the CPO Server, or a user of the EV Charging System. The requirement only asks for the first type of authentication, because the OCPP protocol does not support more fine-grained authentication. The first type of authentication can be implemented for instance by using server certificates within TLS.

This requirement is usually met by using message authentication code (MAC) or a block cipher in authenticated encryption mode, as for requirement SCR.03. These algorithms allow the *Device* to check that a message is sent by someone who has access to the key used for them. Requirement SCR.05 puts restrictions on who can have the key. If pre-shared keys are used, different keys must be used for different roles or hosts. If session keys are used, the protocol used to agree on the session key should check whether the user making the request has a certain role or is in on a certain host.

This requirement does not apply to the Authentication Terminal, for the same reason as for requirement SCR.01.

SCR.06 Non-Repudiation

Device	• Local Controller • EVSE • Authentication Terminal

Minimum Requirements	1. The Device SHALL support non-repudiation for firmware: when it installs firmware, it SHALL be able to prove that the firmware came from the Vendor.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used for non-repudiation. • Functional testing can be used to verify if the mechanisms are indeed implemented. • Penetration tests can be used to ascertain that attackers cannot bypass the non-repudiation mechanisms.

Non-repudiation means that a sender of the firmware should not be able to deny that he sent it. It is normally implemented using digital signatures. A hash value of the firmware is calculated, and signed using public-key cryptography. The private key is kept by the Vendor (see SDR.03). The public key for the validation of the signature can be installed on the *Device* during the manufacturing process. SDR.08 defines Production Security & Credential Provisioning. It is not needed to keep the public key secret. Measures should be taken to make sure the correct key is installed however.

It is not necessary that the Purchaser establishes a Public Key Infrastructure (PKI) at the Central System for this purpose. The Vendor has to store the private firmware signing key as express in the Section 4 Product Lifecycle and Governance.

2.4 System Hardening

The requirements in this section concern hardening of the EV charging system *Devices*. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SHR**.".

SHR.01 Device Hardening

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device SHALL have all unneeded services and applications removed, or disabled if removal is not possible. 2. The Device SHALL not use services or applications for security functions if there are vulnerabilities known for them. 3. The Device SHALL use only communication protocols that are needed to meet the functional requirements.
Recommended Assurance	• Vulnerability scanners can automatically check devices for known vulnerabilities.

	• Carrying out a penetration test can provide further assurance that this requirement is adequately implemented. • If high-impact functions are disabled in the Device code, the Purchaser can request a code review from the Vendor.

Examples of unused services and application that should be removed or disabled are: Testing and debugging applications used for initialization or testing during the production process.

Webservers used as graphical user interfaces (GUIs) or for maintenance purposes if maintenance is normally done through a specialized application.

- FTP servers used during installation.
- Drivers for hardware that is not in the *Device.*
- A telnet service when SSH is also available.
- NTP or DNS servers if these are not used by other devices in the EV Charging System.
- Vulnerabilities are considered known if they are in a public vulnerability database, or if an advisory on them has been published.

Webservers/GUIs are often prone to code injection, buffer overflows and other vulnerabilities, they pose a high risk when directly accessible from a remote connection. The OWASP list [30] provides a good overview of known web vulnerabilities.

SHR.02 Interface Minimization

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device SHALL have any unneeded interfaces and ports removed, or disabled if removal is not possible. In particular, all hardware interfaces that are used for debugging MUST be completely removed after production. 2. The Device SHALL not allow direct remote access to modules apart from the local controller.
Recommended Assurance	• Carrying out a penetration test can provide assurance that this design requirement is adequately implemented.

Redundant and unused ports could include

- USB ports
- Ethernet ports
- Serial ports

Microcontrollers and processors are often equipped with hardware interfaces, such as JTAG, and Serial Wire Debug. These interfaces allow programming or debugging of the respective components and are required for example in the course of production. They should be disabled in operational systems.

SHR.03 Account Hardening

Device	• Local Controller
Minimum Requirements	1. The Device MUST NOT contain active default, guest and anonymous accounts. 2. The Device MUST not allow remote access to root accounts on the Device. 3. The Device SHALL have Vendor-owned accounts removed where feasible.
Awarding Criteria	4. The Device SHOULD support enforcing a password policy that only allows passwords of sufficient complexity.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Carrying out a penetration test can provide further assurance that this design requirement is adequately implemented.

SHR.04 Security-enhancing features

Device	• Local Controller • Authentication Terminal
Awarding Criteria	6. The Device SHOULD deploy security-enhancing features of the underlying platform, implementation language and tool chain when it enhances the Device security.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on which security enhancing features are used. • Functional tests can be used to verify that features are indeed used.

Examples of security-enhancing features are:

- Compiler options that enhance security, such as adding checks to buffer overruns to the code.
- Secured boot process where the boot loader verifies the integrity of the firmware at startup.

- Use of a secure element such as a Hardware Security Module (HSM) and Trusted Platform Modules (TPM).
- Encryption of non-volatile memory.
- Activation of read-out protection enabling functions of a microcontroller. Whitelisting of programs and services to prevent that malware is executed on the system.

The use of processor features that enhance security, such as ARM Trust Zones. Using these features is not needed to meet the security requirements in this document. They can however add an extra layer of defense.

SHR.05 Protection against Physical Manipulations

Minimum Requirements	1. Physical manipulations of the Charge Point SHALL be recognizable. 2. The Charge Point door SHALL provide sufficient protection against physical manipulations. 3. The opening of the Charge Point door SHALL be recognized using suitable means such as sensors. Any opening of the Charge Point door SHALL generate an event in the security log. 4. The removal of any part of Charge Point SHALL generate an event in the security log.
Awarding Criteria	5. The vendor SHOULD provide design evidence ensuring that this requirement is addressed.
Recommended Assurance	• Carrying out a penetration test can provide further assurance that this design requirement is adequately implemented. • Analysis of the penetration test results.

2.5 Resilience

The requirements in this section concern resilience of the EV charging system *Devices* and the communication sent and received by the EV charging system *Devices*. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SRR**.".

SRR.01 Message Validity Verification

Device	• Local Controller • Authentication Terminal

Minimum Requirements	1. The Device SHALL verify the validity of all messages it receives. 6. The Device SHALL reject or drop messages that are invalid or for which the validity cannot be verified.
Recommended Assurance	• It is recommended to carry out fuzzing tests on all interfaces. • The Vendor should provide a detailed documentation of all security tests.

A message is considered valid if it meets all protocol specifications, it makes sense for the *Device*'s configuration, and it meets all requirements the *Device* has on data sizes. Examples of validity checks include checks of syntax, data format, and value ranges. The *Device* should also check if registers or data objects reference by a message exists, and if the data fits into internal buffers allocated for it.

The requirement is valid for all network protocol layers, including the wireless protocols, TCP/IP stack, and application layer protocols.

SRR.02 Fail-Secure Operation

Device	• Local Controller • Authentication Terminal
Minimum Requirements	1. The Device SHALL be fail-secure, i.e., it SHALL be designed to fail in a manner that limits any security compromise of its own operation and security compromise of other devices. 2. The Device SHALL not leak confidential information, such as keys or credentials, on any interface during a failure. 3. The Device SHALL protect the integrity of security critical data during failures. 4. The Device SHALL not allow access controls to be bypassed remotely during failures. 5. The Device SHALL restore availability after software failures as soon as possible.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Carrying out a penetration test can provide further assurance of the design robustness.

Point 5 can be addressed by implementing a watchdog functionality that allows the device to maintain a secured operational state in case of a failure.

Examples for relevant failures are:

- Integrity errors, e.g. of configurations or log files;
- Failures during execution of cryptographic functions;
- Failures during validation of login credentials;
- Failures when entering data (wrong data format, wrong data length, invalid commands etc.).

3 Support for Secure Operation

The requirements in this section concern access control and logging of security events, two services needed to securely operate the EV charging system *Devices*.

3.1 Access Control

The requirements in this section concern access control for the EV charging system *Devices*. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SAR**.".

SAR.01 Access Control

Device	• Local Controller
Minimum Requirements	1. On the WAN interface, the Device SHALL allow to restrict access to certain hosts. 2. On the Maintenance interface, the Device SHALL allow to set access privileges to functions per role. 3. On the Maintenance interface, the Device SHALL only grant access to configuration and firmware update functions if a user's role has the right privileges. 4. The Device SHALL allow new roles to be defined. 5. The Device SHALL allow to assign to each role individual security credentials and keys.
Recommended Assurance	• This requirement is verified in a functional security test. The test should in particular ensure that each role has only the defined and necessary privileges. • Penetration testing can be used to make sure that the access controls cannot be circumvented by for instance privilege escalation.

The requirement on the WAN interface is more limited, because the commonly used OCPP protocol does not support more fine-grained access control.

Separation of different roles is required on the Maintenance interface, to simplify maintenance procedures. It allows for instance to set different privileges for engineers from DSOs and from CPOs.

The requirement does not specify how users are assigned to roles. This can be arranged for instance by having the maintenance tools or the Local Controller itself contact a central authentication server.

SAR.02 User Authentication

Device	• Local Controller
Awarding Criteria	1. The Device SHOULD authenticate the communication parties on the WAN interface using a challenge-response protocol based on either message authentication codes or public-key certificates. 2. The Device SHOULD terminate the connection if the user authentication fails. 3. The Device SHOULD authenticate the communication parties on the Local Maintenance interface. 4. The Device SHOULD support blocking authentication requests, either temporarily or permanently, from an account after a number of failed login attempts. The number of failed login attempts and the time the account is blocked SHOULD be configurable.
Recommended Assurance	• The implementation of user identification can be verified in a functional security test. • Carrying out a penetration test can provide further assurance that this design requirement is adequately implemented.

3.1.1 User Authentication for the Authentication Terminal

One area of special concern is how end-users, that is people who want to charge their car at the Charge Point, authenticate themselves. It is assumed that they authenticate themselves with a token, such as an RFID card. The Authentication Terminal uses the UA interface to read and authenticate the end-user token. Once the end-user token has been successfully authenticated, the Authentication Terminal forwards the end-user token ID to the CPO for management of authorization of access to the EV services.

SCR.07 End User Authentication

Device	• Authentication Terminal
Minimum Requirements	1. The Device MUST support a cryptographic challenge-response authentication protocol to authenticate the end-user token. 2. If the challenge-response protocol is used, the Device SHALL only accept an end-user token ID as valid once the end-user token has been successfully authenticated. 3. The Device MUST support UID identification. 4. The Device MUST support disabling the UID identification mechanism remotely, if this is desired.

	5. If a common master key is used in the authentication protocol that is shared between all Charge Point, the Device SHALL store it in a Secure Access Module.
Awarding Criteria	2. The Device SHOULD rely on an internal Secure Access Module (SAM) to manage keys involved in the authentication protocol.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the authentication protocol. • Functional testing can be used to verify if the authentication protocol is indeed implemented. • Penetration tests can be used to ascertain that attackers cannot bypass the authentication protocol.

The authentication between the AT and the end-user token consists in verifying that this token is valid and has been issued by an authorized party. This allows to use the ID it contains as the identity of the end-user.

The support for UID identification as an authentication mechanism is integrated for supporting legacy tokens. It shall be however possible to disable this mechanism in the future, if it is no longer desired to use UID identification mechanism.

A challenge-response authentication protocol prevents replay attacks.

Cryptographic algorithms and related key lengths used in the authentication protocol needs to be comply with required stated in SPR.01. Also, challenge data used in the authentication protocol needs to be using cryptographic randomness as in SPR.02.

With a challenge response protocol, authentication protocol between the AT and the end-user token can rely on 2 types of keys:

- A diversified symmetric key. End-user tokens keys are obtained from a key derivation algorithm from a common master key and the ID of the token. Compromise of a specific token key only compromise this key, not all token keys. However, Authentication Terminals have to have access and store the common master key. A Secure Access Module (SAM) has to be used to securely store this key and derive tokens keys according to their ID.
- Asymmetric keys. End-user tokens store private keys to authenticate, whereas the Authentication Terminal only has to store a public key. This requires more high-end tokens and can introduce a delay in the duration of the authentication.

For token authentication, the stored credentials on the AT depend on the authentication protocol. It may be a shared symmetric key, or a public key of the token.

Trust in the system also rely on the tamper protection of the end-user token, that prevents unauthorized users to access or modify cryptographic keys and ID stored in the token. Procurement of reliable, recent tokens and whose resistance to attacks has been evaluated, for instance according to Common Criteria standard [19], is recommended.

The implementation of these security mechanisms is typically performed at the application layer, and may vary. Developers shall however ensure that the underlying transport protocol is compatible with security mechanisms such as challenge-response.

Typically, most RFID applications on cards should be compatible either with ISO 14443 or ISO 15693 [40]. Other kinds of tokens may be used, such as smartphones, in what case standards such as ISO 18092 [41] may be used. These standards do not define security mechanisms, but allow the use of security mechanisms such as challenge-response, through applicative protocols such as ISO 7816-3 and 7816-4 [42] [42](for smartcards) or proprietary mechanisms such as Mifare [43], FeliCa [44], Calypso [45] or Cipurse [46].

Developers shall verify whether the cryptographic mechanisms offered by the authentication protocols comply with SPR.01.

3.2 Logging

The requirements in this section concern logging of events. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SLR**.".

SLR.01 Logging Security Events

Device	• Local Controller
Minimum Requirements	1. The Device SHALL log security events in a locally stored log. 2. The Device SHALL take measures to prevent that attackers can modify, delete or overwrite the security log to hide their traces. 3. The Device SHALL support automatically sending log events to a central logging server or SIEM. 4. The Device SHALL support synchronization with a centrally maintained time.
Awarding Criteria	5. The Device should allow remote monitoring of information about the device status such as processor and memory usage. 6. The Device should store for each security event at least the interface, the event type, a time stamp, and the user, role, or process causing the event.
Recommended Assurance	• The implementation of logging mechanisms can be verified in a functional security test. • Carrying out a penetration test can provide further assurance that attackers cannot bypass detection mechanisms or modify the security log.

In the requirements below security events are any events relevant to the secure operation of the *Device.* Security events include at least the following:

- User Activities:
  - Successful logins
  - Failed login attempts
- Changes of security credentials
  - Unauthorized file access
- Possible signs of attacks:
  - Resource exhaustion (DoS)
  - Messages whose integrity could not be verified
  - Invalid messages
  - Attempted replay attacks
  - Alarms on physical manipulations
- Updates or changes:
  - Firmware Updates or patches
  - Configuration Changes

Common methods to export security events to a central logging server are syslog and SNMP. Syslog allows integration with many different SIEM solutions. Time synchronization is required to allow logs events from different devices to be correlated. Different technologies are available for time synchronization, such as NTP and GPS.

SLR.02 Logging Security Events

Device	• Authentication Terminal
Minimum Requirements	1. The Device SHALL send the log security events to the Local Controller.
Awarding Criteria	• The Device should send to the Local Controller for each security event at least the interface, the event type, a time stamp, and the user, role, or process causing the event.
Recommended Assurance	• The implementation of logging mechanisms can be verified in a functional security test. • Carrying out a penetration test can provide further assurance that attackers cannot bypass detection mechanisms or modify the security log.

In the requirements below security events are any events relevant to the secure operation of the *Device.* Security events include at least the following:

- User Activities:
    - Successful logins
    - Failed login attempts
- Changes of security credentials
    - Unauthorized file access
- Possible signs of attacks:
    - Invalid transaction messages
- Updates or changes:
    - Firmware Updates or patches
    - Configuration Changes

4 Product Lifecycle and Governance

The requirements in this section concern the processes used for developing, manufacturing, and provisioning of the EV charging system *Devices* in a secure way. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SDR**.".

There will be no recommendation regarding quality assurance for the requirements in this section. It is recommended that the Purchaser asks for documentation to verify the implementation of the requirements.

All requirements hold for the complete contractually agreed lifecycle of the EV charging system *Devices*. All requirements apply to the Vendor and suppliers. This includes in particular Third-Party Suppliers.

SDR.01 Information Security Management System

Minimum Requirements	1. The Vendor SHALL implement an information security management system (ISMS) the scope of which includes at least all systems used to develop, test, manufacture and provision the Devices and any software and hardware tools needed for the maintenance of the Device.
Awarding Criteria	2. The Vendor SHOULD have regular audits of the ISMS performed by an accredited external auditor. 3. The Vendors SHOULD provide a proof of the audit to the Purchaser on request. 4. The Vendor SHOULD obtain an ISO 27001 certification for the ISMS. 5. The Vendor SHOULD make a proof of the certificate available on request. 6. The Vendors SHOULD share their security policies with the Purchaser.

Quality assurance certification schemes such as the ISO 9001 are not sufficient to meet this requirement.

SDR.02 Configuration Management System

Minimum Requirements	1. The Vendor SHALL employ a configuration management system for the administration of (changes of) hardware configurations and source code of devices. 2. The Vendor SHALL ensure that the configuration management system stores for each change an explanation, the author, the parts changed, and the time at which it was made.

Awarding Criteria	3. The Vendor SHOULD allow the purchaser to audit the configuration management system.

SDR.03 Secured Versioning

Minimum Requirements	1. The Vendor SHALL ensure that all released versions of hardware and firmware of the Device are uniquely identifiable. 2. The Vendor SHALL provide to the Purchaser a cryptographic hash value for each firmware version. 3. The Vendor SHALL be able to reproduce released versions within the contractually agreed product lifecycle, with traceability provided by the hash value(s) as identifier(s). 4. The Vendor SHALL version exchangeable hardware modules separately. 5. The Vendor SHALL digitally sign each firmware update supplied to the Purchaser. 6. The Vendor SHALL protect the firmware signing keys as highly confidential data. 7. The Vendor SHALL report it to the Purchaser if a firmware signing key is compromised.

SPR.01 gives references for allowed cryptographic hash functions, and digital signing algorithms.

The ISMS required by SDR.01 is normally used to determine the measures needed to protect the firmware signing key. Point 6 of this requirement means that a compromise of the confidentiality of the key should be treated as a high impact event in the ISMS.

SDR.04 Vulnerability Handling Process

Minimum Requirements	1. The Vendor SHALL have an established and documented process to handle vulnerabilities. 2. The Vendor SHALL monitor information sources on vulnerabilities to determine if it has been affected. 3. The Vendor SHALL address vulnerabilities found by the Vendor itself, the Purchaser or system integrator, or external security researchers. 4. The Vendor SHALL disclose to the Purchaser all known vulnerabilities on the Device as soon as possible.

	5. The Vendor SHALL communicate vulnerabilities to the Purchaser in a secure manner. 6. The Vendor SHALL issue a recommendation on how to mitigate a vulnerability as soon as possible. 7. The Vendor SHALL evaluate the criticality of a vulnerability using established standards (such as CVSS [36]). 8. The Vendor SHALL prioritize fixing vulnerabilities based on the potential impact to the Purchaser.

Standards are available to objectively assess the impact of vulnerabilities, such as CVSS [36]. These can be used as an aid to prioritize fixing vulnerabilities. It is however recommended that the Vendor also takes into account the specific design of the Device, and how it is used by the Purchaser, when assessing the potential impact.

SDR.05 Security Updates and Patching

Minimum Requirements	1. The Vendor SHALL provide security updates or patches for the Device to fix high impact vulnerabilities found during the Device's lifecycle. 2. The Vendor SHALL test all security updates and patches prior to deployment.
Awarding Criteria	3. The Vendor SHOULD provide documentation that all security patches were tested and validated prior to deployment. 4. The Vendor SHOULD provide tools enabling batch updating of Devices. 5. The Vendor SHOULD release a patch or firmware update for a vulnerability no more than three months after it was reported to the Vendor.

The Vendor is allowed to leave vulnerabilities with a low impact unpatched. The impact is defined after a risk analysis of the vulnerability as specified in SDR.06. Of course it is not recommended to do so. Low impact vulnerabilities should always be disclosed to the Purchaser by requirement SDR.04.

SDR.06 Security Training and Awareness

Minimum Requirements	1. The Vendor SHALL provide security training for the personnel.

	2. The Vendor SHALL be able to document that the necessary knowledge to securely develop and securely produce products is in place. 3. The Vendor SHALL name a technical expert responsible for security-related matters who acts as contact person for the Purchaser. 4. The Vendor SHALL conduct a risk analysis of the firmware design and the corresponding system architecture.
Awarding Criteria	5. The Vendor SHOULD provide documented professional experience in the area of IT security or a security certification, e.g., CISSP or CISM.

SDR.07 Production Security and Credential Provisioning

Minimum Requirements	1. The Vendor SHALL ensure secure provisioning of cryptographic keys, passwords and initial security credentials during the manufacturing process. 2. The Vendor SHALL ensure a secure production area to ensure the secure initial provisioning of credentials and cryptographic keys to the device. 3. The Vendor SHALL establish a secure hand-over process of the provisioned information to the central systems of the Purchaser.

Initial security credentials include passwords.

5 Assurance

The requirements in this section concerns measures the Vendor should take to make sure the EV charging system *Devices* will work securely. Requirements are grouped into different items. Each item has a unique identifier with prefix "**SUR**.".

SUR.01 Design Evidence

Minimum Requirements	1. The Vendor SHALL document all interfaces of the Device, including the protocols and services used on each interface. 2. The Vendor SHALL provide design evidence that sufficient reserves are available to update security functionality to meet requirement SFR.01. 3. The Vendor SHALL provide design evidence that only cryptographic algorithms, protocols, and parameters allowed by SPR.01 are used for security functions, including a description of which algorithms, protocols, and parameters are used for which functions. 4. The Vendor SHALL provide design evidence that cryptographic random number generation is implemented according to requirement SPR.02, including a description of which random number generator is used. 5. The Vendor SHALL provide design evidence of the authentication protocol required in for SCR.01. 6. The Vendor SHALL provide design evidence that firmware authenticity is protected as required in SCR.02, including a step-by-step description of the firmware update process. 7. The Vendor SHALL provide design evidence that unused interfaces are disabled or removed to meet requirement SHR.02. 8. If interfaces or services or disabled and not removed, the Vendor SHALL provide information on how they have been disabled. 9. If security-enhancing features as described in requirements SHR.03 are used, the SHALL provide design evidence on how they are used. 10. The Vendor SHALL provide design evidence on how the Device has been made fail-secure to meet requirement SRR.02, including a list of all relevant failure types and their countermeasures. 11. The Vendor SHALL provide design evidence that user authentication is implemented as required in SAR.01. 12. The Vendor SHALL provide design evidence that security logging is implemented as required in SLR.01. 13. The Vendor SHALL provide design evidence at a level of detail that makes it easy to verify that the security requirements are

	implemented, and to test that they are implemented on the Device as described. 14. The Vendor SHALL allow verification of the design evidence by an independent third party selected by the Purchaser.

This requirement stresses that the Vendor provides the Purchaser design evidence. **Design evidence** consists of documents produced during the design and development processes that explain how the security requirements have been implemented on the Device. The requirements in this document are formulated in a technology independent manner. The Vendor has different options to implement them. To allow the Purchaser to verify that the requirements are implemented correctly, it is important that they understand which option was chosen.

If design evidence is sensitive from a security or competitive viewpoint, the Vendor can supply it under an NDA, as long as the NDA allows for verification of the design evidence by the Purchaser or an independent third party.

SUR.02 Security Testing

Minimum Requirements	1. The Vendor SHALL perform tests to verify that all the security requirements in this document have been implemented correctly. 2. These Vendor SHALL test the complete functional scope of the Device, including the communication chain between the Device and all connected field devices and the central systems. 3. The Vendor SHALL test both regularly used as well as rarely used functionalities of the Device. 4. The Vendor SHALL document the concepts and details of the security tests in a comprehensible way. 5. The Vendor SHALL use vulnerability scanners to test each released firmware version on known vulnerabilities. 6. The Vendor SHALL allow the Purchaser to contract an independent test lab to perform a security tests on the Device.
Awarding Criteria	7. The Vendor SHOULD conduct robustness tests, such as fuzzing or flooding, on all protocols used by the device both on the application layer and on lower protocol layers. 8. The Vendor SHOULD conduct design and code reviews and provide the results to the Purchaser.

Examples of security tests to verify the requirements are given for each requirement under quality assurance.

SUR.03 Secure Coding Practices

Awarding Criteria	1. The Vendor SHOULD establish and enforce secure coding practices for the development of the Device following best practices. 2. The Vendor SHOULD establish an internal code review process that takes security into account. 3. The Vendor SHOULD use automated code analysis tools to find security vulnerabilities.

Examples of secure coding practices are the SEI CERT coding standards [34], available for different languages, and the MISRA C software development guidelines for embedded systems. [20]

SUR.04 Secure Audit Collaboration

Minimum Requirement	1. If the Purchaser desires to perform an additional security audit, the Vendor SHALL collaborate with an external testing party named by the Purchaser

6 Requirements for CPO and DSO Communication

The requirements in this section specify the measures that should be taken to secure communication between the CPO and DSO Servers. The requirement numbering is consistent with the requirements for the Charge Point in Section 2. See the requirements there for additional comments on the possible implementation.

SCR.01.CPO Confidentiality

Minimum Requirements	1. The CPO system SHALL protect the confidentiality of communication by encrypting it using a protocol allowed by SPR.01 over the CPO interface. 2. The CPO system SHALL protect the confidentiality of communication by encrypting it using a protocol allowed by SPR.01 over the WAN interface.
Recommended Assurance	• This requirement is verified in a functional security test. The test should in particular ensure that the allowed cryptographic algorithms are supported and that disallowed algorithms are rejected.

SCR.01.DSO Confidentiality

Minimum Requirements	1. The DSO system SHALL protect the confidentiality of communication by encrypting it using a protocol allowed by SPR.01 over the CPO interface.
Recommended Assurance	• This requirement is verified in a functional security test. The test should in particular ensure that the allowed cryptographic algorithms are supported and that disallowed algorithms are rejected.

SCR.02.CPO Message Integrity

Minimum Requirements	1. The CPO system SHALL verify the integrity of application layer messages received, using a message authentication algorithm allowed by SPR.01 over the CPO interface. 2. If the CPO system detects that a message has been modified or if it cannot verify the integrity of the message over the CPO interface, it SHALL reject or drop the message. 3. The CPO system SHALL allow parties it communicates; to verify the integrity of application layer messages it sends by using a message authentication algorithm allowed by SPR.01 over the CPO interface. 4. The CPO system SHALL verify the integrity of application layer messages received, using a message authentication algorithm allowed by SPR.01 over the WAN interface. 5. If the CPO system detects that a message has been modified or if it cannot verify the integrity of the message over the WAN interface, it SHALL reject or drop the message. 6. The CPO system SHALL allow parties it communicates; to verify the integrity of application layer messages it sends by using a message authentication algorithm allowed by SPR.01 over the WAN interface.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Functional tests can be used to verify that the CPO system supports the required functionality. • Carrying out a penetration test can be used to determine if the CPO system verifies message integrity under all conditions.

SCR.02.DSO Message Integrity

Minimum Requirements	1. The DSO system SHALL verify the integrity of application layer messages received, using a message authentication algorithm allowed by SPR.01 over the CPO interface. 2. If the DSO system detects that a message has been modified or if it cannot verify the integrity of the message over the CPO interface, it SHALL reject or drop the message. 3. The DSO system SHALL allow parties it communicates; to verify the integrity of application layer messages it sends by using a message authentication algorithm allowed by SPR.01.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor. • Functional tests can be used to verify that the DSO system supports the required functionality. • Carrying out a penetration test can be used to determine if the DSO system verifies message integrity under all conditions.

SCR.04.CPO Message Freshness

Minimum Requirements	1. The DSO system SHALL be able to detect replay attacks over the CPO interface. 2. If the DSO system detects that a message is replayed, it MUST reject or drop the message.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used to protect against replay attacks. • Functional testing can be used to verify if the mechanisms are indeed implemented.

SCR.04.DSO Message Freshness

Minimum Requirements	1. The DSO system SHALL be able to detect replay attacks over the CPO interface. 2. If the DSO system detects that a message is replayed, it MUST reject or drop the message.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used to protect against replay attacks. • Functional testing can be used to verify if the mechanisms are indeed implemented.

SCR.05.CPO Message Authentication

Minimum Requirements	1. The CPO system SHALL be able to determine that the source of a sensor reading request or control command is a specific host in the EV Charging system. 2. The CPO system SHALL be able to determine that the source of a OCCP message is the DSO system.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used for message authentication. • Functional testing can be used to verify if the mechanisms are indeed implemented. • Penetration tests can be used to ascertain that attackers cannot bypass the authentication mechanisms.

SCR.05.DSO Message Authentication

Minimum Requirements	1. The DSO system SHALL be able to determine that the source of a OSCP message is the CPO system.
Recommended Assurance	• Analysis of the design documentation provided by the Vendor on the mechanisms used for message authentication. • Functional testing can be used to verify if the mechanisms are indeed implemented. • Penetration tests can be used to ascertain that attackers cannot bypass the authentication mechanisms.

SRR.01.CPO Message Validity Verification

Minimum Requirements	1. The CPO system SHALL verify the validity of all messages it receives. 2. The CPO system SHALL reject or drop messages that are invalid or for which the validity cannot be verified.
Recommended Assurance	• It is recommended to carry out fuzzing tests on all interfaces. • The Vendor should provide a detailed documentation of all security tests.

SRR.01.DSO Message Validity Verification

Minimum Requirements	1. The DSO system SHALL verify the validity of all messages it receives. 2. The DSO system SHALL reject or drop messages that are invalid or for which the validity cannot be verified.
Recommended Assurance	• It is recommended to carry out fuzzing tests on all interfaces. • The Vendor should provide a detailed documentation of all security tests.

7 Glossary

This glossary serves as inventory of technical terms and abbreviations used in the document. For detailed background information on cryptographic primitives or testing procedures we refer to the referenced literature.

AES	Advanced Encryption Standard. Original name for this block cipher was Rijndael named after its designers Vincent Rijmen and Joan Daemen.
Application layer	OSI-Layer 5-7.
Authentication	When speaking about authentication one should distinguish between user authentication (e.g., sender/receiver) and message authentication.
Block cipher	Cryptographic primitive to encrypt/decrypt messages of fixed block length. Example: AES encrypts blocks of 128 bits (16 bytes) at a time.
Block cipher Mode of Operation	A mode of operation specifies how the message blocks are processed by the block cipher. Using a block cipher in CBC or CTR mode provides encryption only whereas using a block cipher in CCM or GCM mode encrypts the plaintext and produces a message authentication tag for the ciphertext.
Certificate	A digital certificate authenticates a public key or entity. See also Public-Key Infrastructure.
Challenge-response authentication	Mechanism to prove an entity's identity to another entity. (e.g. username/password)
Confidentiality	Only authorized entities may access confidential data. To protect data from unauthorized access it can be encrypted. Then only entities with access to the secret keys can access the data after decrypting it.
Cryptographic hash function	Cryptographic hash functions should behave as one-way functions. They must be preimage resistant, 2nd preimage resistant, and collision-resistant. Changes in the input must produce explicitly different results in the output. Example: SHA-256. See also ENISA [12].
Cryptographic protocol	A protocol used for security functions, such as authentication protecting confidentiality or integrity.

Cryptography	The ENISA Algorithms, Key Sizes and Parameters Report [12] provides an overview of the current state of the art.
Data Integrity	See Integrity and Message authentication.
Design Evidence	Documents produced during the design and development processes that explain how the security requirements have been implemented on the *Device*.
Digital Signature	Authenticates the sender. In practice digital signatures are implemented using elliptic curves (EC). See standards such as [14][18] and [25][31] for the implementation of the Elliptic Curve Digital Signature Algorithm (ECDSA).
EC	Elliptic Curve. See also ENISA [12].
ECDSA	Elliptic Curve Digital Signature Algorithm.
Encryption	Using a cryptographic scheme the message is mapped to a random-looking undecipherable string (ciphertext). Decryption reverses the encryption process and can only be performed with the corresponding decryption key. This decryption key is either the same as the encryption key (symmetric cryptography) or the private key in a public-key cryptosystem. The confidentiality of the message can be guaranteed only while the keys are kept secret.
End-User Token	An End-User Token is a device that includes a chip to store the Unique Identification Number of the user. It is used combined with a wireless technology such as RFID to authenticate the EV User.
ENISA	European Union Agency for Network and Information Security.
EPRI	Electric Power Research Institute.
Fuzzing Test	A fuzzing test provides quality control of software used for secure network communication. A fuzzing test generates a high volume of mostly random data including malformated messages and observes the reaction of the device/system under test. More information on fuzzing is provided in [26][32].
GPRS	General Packet Radio Service.
GPS	Global Positioning System.
Hash function	Function that maps a message to a bit string of fixed length (hash value). See also cryptographic hash function.

Hash value	Output of a (cryptographic) hash function. The length is fixed in the specs of the hash function.
ICS	Industrial Control System.
IETF	Internet Engineering Task Force.
Integrity	Data cannot be altered without authorization. See also message authentication.
ISO 27001	ISO standard for information security. Current version at the time of writing: ISO27001:2013.
Key material	The term 'key material' includes all cryptographic keys. Examples: master key, symmetric session keys, private and public keys (public-key cryptography).
LAN	Local Area Network.
LDAP	Lightweight Directory Access Protocol.
Maintenance interface	See Deliverable D1.1 [1] on the reference architecture.
MAC	Message authentication code. Provides data integrity. Examples: CMAC, GMAC. See also ENISA [12].
Message authentication	Messages should be protected against unauthorized modifications. The message should always be sent together with an authentication tag providing its authenticity. Such an authentication tag can be the second output of an authenticated cipher such as AES-CCM or AES-GCM or a message authentication code.
Message Validity	A message is considered valid if it meets all protocol specifications, it makes sense for the *Device's* configuration, and it meets all requirements the *Device* has on data sizes.
NESCOR	National Electric Sector Cybersecurity Organization Resource. Program issued by the US organization EPRI. See [27][33].
NIST	National Institute of Standards and Technology.
Nonce	A nonce is a unique randomly generated string which can be used exactly once. Attachment of a nonce helps to prevent replay attacks.
NTP	Network Time Protocol.

OSI	Open Systems Interconnection. Reference model for network communications.
Password authentication	The user proves his/her identity using a password or PIN.
Penetration test	For a guideline refer to the EPRI program NESCOR, specifically the "AMI Penetration Test Plan".
Product lifecycle	The product lifecycle spans all stages of a product: starting from the design through the development and production to delivery and decommissioning. The Purchaser and Vendor should agree on the length of the product lifecycle.
Public-key cryptography	Cryptographic scheme where a public key is published and henceforth can be used for encryption of messages or verification of digital signatures. Each public key has a counterpart, the corresponding private key. This key must be kept secret and is used for decryption or digital signing of messages. Public-key primitives have a high computational complexity for encryption and therefore are mostly used as part of a hybrid encryption scheme where the public key is used to communicate a common symmetric session key under which all further communication is encrypted. Certificates administered by a public-key infrastructure are used to establish the authenticity of the public key. See also ENISA [12]. The most popular public-key encryption scheme is RSA. Digital signatures can be generated most efficiently with elliptic-curve based (EC) mechanisms.
Public-key infrastructure	System to generate, administer, and revoke certificates.
Replay attack	The attacker observes and captures data during a session with the intention of resending it later and thus impersonating one communication partner.
RFC	Requests for Comments. Published by the IETF.
Robustness test	A robustness test provides quality control by checking the design stability/robustness of the system. The tests check in particular the fault tolerance of the system.

RSA	Public-key cryptosystem named after its inventors Rivest, Shamir, and Adleman.
Security Event	Any event relevant to the secure operation of the *Device*.
Security Function	Any function on the *Device* that is needed for it to be operated securely, including access control, authentication, and encryption.
Session key	Symmetric key with a limited lifetime.
Symmetric cryptography	Sender and receiver hold the same key. Examples for symmetric primitives are block ciphers or MACs.
User Authentication	Verification of the identity of the communication partners (e.g., user on the local controller). Moreover, verification that the communication partners are still alive throughout a session. See also password authentication and user authentication.
WAN	Wide Area Network.
WAN Interface	Remote connection to Central System. See Deliverable D1.1 [1] for the Reference Architecture.

8 References

[1] European Network for Cyber Security. Reference Architecture for Secure EV Charging Systems. Deliverable D1.1 in the EV Charging Project. Final Version, April 2016.

[2] European Network for Cyber Security. Reference Threat Assessment for Secure EV Charging Systems. Deliverable D1.2 in the EV Charging Project. Final Version, April 2016.

[3] European Network for Cyber Security. Mapping of OCPP with analysis against the Reference Requirement for Secure EV Charging System. Final Version, April 2016

[4] BDEW Bundesverband der Energie- und Wasserwirtschaft e.V., Anforderungen an Sichere Steuerungs und Telekommunikationssysteme (Requirements for Secure Control and Telecommunication Systems), v.01, 2008 (English and German).

[5] Department of Homeland Security (DHS). Cyber Security Procurement Language for Control Systems. September 2009.

[6] North American Electric Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) Standards. http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx (last accessed on 17 January 2016)

[7] IEG 60068-2-75:2014, Environmental testing - Part 2-75: Tests - Test Eh: Hammer tests

[8] IEC Technical Committee 23, "IEC 62196 ed2.0: Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles," Geneva, 2011.

[9] Open Charge Point Protocol SOAP 1.6, OCPP-S 1.6 Specification, 2015

[10] IEC 62443 and ISA99, Industrial Automation and Control Systems Security Standards.

[11] IEC 62351. Power systems management and associated information exchange – Data and communications security. Parts 1-8.

[12] ENISA European Network and Information Security Agency, Algorithms, key size and parameters report 2014, 2014. (last accessed on 17 January 2016)

[13] reserved

[14] Internet Engineering Task Force. RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2), 2014. https://tools.ietf.org/rfc/rfc7296.txt (last accessed on 17 January 2016)

[15] Internet Engineering Task Force. RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2, 2008. http://www.ietf.org/rfc/rfc5246.txt (last accessed on 17 January 2016)

[16] Bundesamt für Sicherheit in der Informationstechnik. TR-02102-3: Kryptographische Verfahren: Empfehlungen und Schlüssellängen Teil 3 – Verwendung von Internet Protocol Security (IPsec) und Internet Key Exchange (IKEv2). Bonn, Germany. Version 2015-01.

[17] Internet Engineering Task Force. RFC 5289: TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), 2008. http://www.ietf.org/rfc/rfc5289.txt (last accessed on 17 January 2016)

[18] National Institute of Standards and Technology. Special Publication 800-57 Part 1 Rev. 3, Recommendation for Key Management, July 2012.

[19] Common Criteria Evaluation methodology, September 2012, Version 3.1 revision 4, ref. CCMB-2012-09-004

https://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R4.pdf

[20] MISRA C software development guidelines for embedded systems, http://www.misra.org.uk/

[21] National Institute of Standards and Technology. Special Publication 800-38C: Recommendation for block cipher modes of operation. The CCM mode for authentication and confidentiality (including updates as of 07-20-2007). 2007.

[22] National Institute of Standards and Technology. Special Publication 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. November 2007.

[23] National Institute of Standards and Technology. Cryptographic Algorithm Validation Program. http://csrc.nist.gov/groups/STM/cavp/ (last accessed on 17 January 2016)

[24] National Institute of Standards and Technology. Annex C: Approved Random Number Generators for FIPS PUB 140-2 [25], February 2012.

[25] National Institute of Standards and Technology. FIPS PUB 140-2, Security Requirements for Cryptographic Modules, May 2001.

[26] Bundesamt für Sicherheit in der Informationstechnik: Anwendungshinweise und Interpretationen zum Schema, AIS 20, Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, Version 3.0, Bonn, Germany, May 2013. (in German)

[27] Bundesamt für Sicherheit in der Informationstechnik: Anwendungshinweise und Interpretationen zum Schema, AIS 31, Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, Version 3.0, Bonn, Germany, May 2013. (in German)

[28] Bundesamt für Sicherheit in der Informationstechnik. TR-02102-1: Kryptographische Verfahren: Empfehlungen und Schlüssellängen. Bonn, Germany. Version 2015-01. (in German)

[29] Internet Engineering Task Force. PKCS #5: Password-Based Cryptography Specification Version 2.0, 2000. http://tools.ietf.org/rfc/rfc2898.txt (last accessed on 17 January 2016)

[30] Open Web Application Security Project. https://www.owasp.org/index.php/Data_Validation (last accessed on 17 January 2016)

[31] Bundesamt für Sicherheit in der Informationstechnik. TR-03116, Part 3, Kryptographische Vorgaben für Projekte der Bundesregierung – Intelligente Messsysteme. In German. Annually adapted. Bonn, Deutschland, Date: 2014.

[32] Ari Takanen, Jared DeMott, and Charlie Miller. Fuzzing for Software Security Testing and Quality Assurance (1 ed.). Artech House, Inc., Norwood, MA, USA, 2008.

[33] Electric Power Research Institute. National Electric Sector Cybersecurity Organization Resource. http://www.smartgrid.epri.com/nescor.aspx (last accessed on 17 January 2016)

[34] SEI CERT Coding Standards, https://www.securecoding.cert.org/confluence/display/seccode/SEI+CERT+Coding+Standards

[35] National Institute of Standards and Technology. Special Publication 800-22 Rev. 1a, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, April 2010.

[36] National Institute of Standards and Technology. NISTIR 7946. CVSS Implementation Guidance. April 2014.

[37] BSI, „TR-02102-1 v2015-1: Kryptographische Verfahren: Empfehlungen und Schlüssellängen," 2015

[38] ANSSI, „Mécanismes cryptographiques - Règles et recommandations, Rev 2.03," 2014.

[39] ISO/IEC 14443 Identification Cards – Proximity Integrated Circuit Cards

[40] ISO/IEC 15693 Identification Cards – Contactless Integrated Circuit Cards – Vicinity Cards

[41] ISO/IEC 18092 Near-Field Communication Interface

[42] ISO/IEC 7816-3 Identification cards - Integrated circuit cards - Part 3: cards with contacts - Electrical interface and transmission protocols

ISO/IEC 7816-4 Identification cards – Integrated circuit cards – Part 4: organization, security and commands for interchange

[43] Mifare (NXP) - https://www.mifare.net/en/

[44] FeliCa (Sony) - http://www.sony.net/Products/felica/

[45] Calypso (Innovatron) – http://www.innovatron.fr/CalypsoFuncSpecification.pdf

[46] Cipurse (OSPT alliance) - http://www.osptalliance.org/the_standard