Improved security for OCPP 1.6-J

edition 2 FINAL, 2020-03-31

OCA white paper:

Improved security for OCPP 1.6-J.

Relevant for OCPP 1.6-J (JSON over WebSockets)

Version History

VERSION	DATE	AUTHOR	DESCRIPTION
1.2 Edition 2	2020-03-31	Paul Klapwijk (*OCA*) Milan Jansen (*OCA*) Robert de Leeuw (*ihomer*)	Edition 2, based on the security fixes in the OCPP 2.0.1 specification
1.0	2018-11-20	Robert de Leeuw (_IHomer)	Final release after last rework check

1. Scope

This white paper describes how the security enhancements, introduced in OCPP 2.0, can be used, on top of OCPP 1.6-J, in a standardized way.

The security part of OCPP 2.0 was developed to strengthen and mature the future development and standardization of OCPP. It is based amongst others on the end-to-end security design by LaQuSo [11]. Security requirements are included, on security measures for both Charge Point and Central System, to help developers build a secure OCPP implementation.

This document contains the following security improvements:

- Secure connection setup
- Security events/logging
- Secure firmware update

1.1. Edition 2

This document is the Edition 2 of "Improved security for OCPP 1.6-J" white paper.

This release is based on OCPP 2.0.1. The first version of this document was based on OCPP 2.0.

After the release of OCPP 2.0, some issues were found in OCPP 2.0. Some of these issues could not be fixed issuing errata to the specification text only, as has been done with OCPP 1.6, but required changes to the protocol's machine-readable schema definition files that cannot be backward compatible. These fixes also impacted this document.

All the fixes to the security parts of OCPP 2.0.1 have been merged into edition 2 of this document.

For the differences between the two versions see the changelog

Edition 2 of this document replaces the original version. OCA advises implementers of OCPP 1.6-J to no longer implement the first version of this document and only use edition 2 going forward.

As a rule, existing numbered requirements are only updated or removed, previously used requirements numbers are never reused for a totally different requirement.

Any mentions of "OCPP 2.0" refers to revision 2.0.1 unless specifically stated otherwise.

1.2. Security Objectives

This section is informative.

OCPP security has been designed to meet the following security objectives:

1. To allow the creation of a secure communication channel between the Central System and Charge Point. The integrity and confidentiality of messages on this channel should be protected with strong cryptographic measures.

2. To provide mutual authentication between the Charge Point and the Central System. Both parties should be able to identify who they are communicating with.
3. To provide a secure firmware update process by allowing the Charge Point to check the source and the integrity of firmware images, and by allowing non-repudiation of these images.
4. To allow logging of security events to facilitate monitoring the security of the smart charging system.

1.3. Design Considerations

This section is informative.

This document was designed to fit into the approach taken in OCPP. Standard web technologies are used whenever possible to allow cost-effective implementations using available web libraries and software. No application layer security measures are included. Based on these considerations, OCPP security is based on TLS and public key cryptography using X.509 certificates. Because the Central System usually acts as the server, different users or role-based access control on the Charge Point are not implemented in this standard. To mitigate this, it is recommended to implement access control on the Central System. To make sure the mechanisms implemented there cannot be bypassed, OCPP should not be used by qualified personnel performing maintenance to Charge Points locally at the Charge Point, as other protocols may be used for local maintenance purposes.

1.4. OCPP-J Only

This section is informative.

This document is for OCPP 1.6-J (JSON over WebSockets) only, OCPP-S (SOAP) is NOT supported. This document was started, as it is seen as a simple step to port OCPP 2.0 security to OCPP 1.6. But as OCPP 2.0/2.0.1 only supports JSON over WebSockets (not SOAP), this document is also written for OCPP 1.6-J only. Adding SOAP to this document would have taken a lot of work and review by security experts.

1.5. General documentation remarks

This section is informative.

This document is based on OCPP 2.0.1. To help developers that are implementing both 1.6J security improvement and OCPP 2.0.1, we have kept the Use Case numbering from OCPP 2.0.1. So when implementing for example Use Case N01, it is the same use case in this document as in the 2.0.1 specification.

1.6. Conventions

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 RFC2119 [13], subject to the following additional clarification clause:

The phrase "valid reasons in particular circumstances" relating to the usage of the terms "SHOULD", "SHOULD NOT", "RECOMMENDED", and "NOT RECOMMENDED" is to be taken to mean technically valid reasons, such as the absence of necessary hardware to support a function from a Charge Point design: for the purposes of this specification it specifically excludes decisions made on commercial, or other non-technical grounds, such as cost of implementation, or likelihood of use.

1.7. References

Table 1. References

REFERENCE	DESCRIPTION
[1]	ENISA European Network and Information Security Agency, Algorithms, key size and parameters report 2014, 2014. (last accessed on 17 January 2016) https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014
[2]	Cooper, D., et al., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Internet Engineering Task Force, Request for Comments 5280, May 2008, http://www.ietf.org/rfc/rfc5280.txt
[3]	Dierks, T. and Rescorla, E., The Transport Layer Security (TLS) Protocol Version 1.2, Internet Engineering Task Force, Request for Comments 5246, August 2008, http://www.ietf.org/rfc/rfc5246.txt
[4]	Hollenbeck, S., "Transport Layer Security Protocol Compression Methods", RFC 3749, May 2004. https://www.ietf.org/rfc/rfc3749.txt
[5]	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) https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_pdf.html
[6]	Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, September 2005. https://www.ietf.org/rfc/rfc4210.txt
[7]	National Institute of Standards and Technology. Special Publication 800-57 Part 1 Rev. 4, Recommendation for Key Management. January 2016. https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final
[8]	RFC 2617. HTTP Authentication: Basic and Digest Access Authentication. https://www.ietf.org/rfc/rfc2617.txt
[9]	RFC 5280. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. https://www.ietf.org/rfc/rfc5280.txt
[10]	OCPP 1.6. Interface description between Charge Point and Central System. October 2015. http://www.openchargealliance.org/downloads/
[11]	Eekelen, M. van, Poll, E., Hubbers, E., Vieira, B., Broek, F. van den: An end-to-end security design for smart EV-charging for Enexis and ElaadNL by LaQuSo1. December 2, 2014. https://www.elaad.nl/smart-charging-end2end-security-design/
[12]	RFC 2818. HTTP Over TLS. https://tools.ietf.org/html/rfc2818
[13]	Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. March 1997. http://www.ietf.org/rfc/rfc2119.txt
[14]	RFC 2986. PKCS #10: Certification Request Syntax Specification, Version 1.7. https://www.ietf.org/rfc/rfc2986.txt

2. Secure connection setup

2.1. Security Profiles

This section defines the different OCPP security profiles and their requirement. This White Paper supports three security profiles:
The table below shows which security measures are used by which profile.

Table 2. Overview of OCPP security profiles

PROFILE	CHARGE POINT AUTHENTICATION	CENTRAL SYSTEM AUTHENTICATION	COMMUNICATION SECURITY
1. Unsecured Transport with Basic Authentication	HTTP Basic Authentication	-	-
2. TLS with Basic Authentication	HTTP Basic Authentication	TLS authentication using certificate	Transport Layer Security (TLS)
3. TLS with Client Side Certificates	TLS authentication using certificate	TLS authentication using certificate	Transport Layer Security (TLS)

- The Unsecured Transport with Basic Authentication Profile does not include authentication for the Central System, or measures to set up a secure communication channel. Therefore, it should only be used in trusted networks, for instance in networks where there is a VPN between the Central System and the Charge Point. For field operation it is highly recommended to use a security profile with TLS.

2.2. Generic Security Profile requirements

Table 3. Generic Security Profile requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.001		The Charge Point and Central System SHALL only use one security profile at a time
A00.FR.002	If the Charge Point tries to connect with a different profile than the Central System is using	The Central System SHALL reject the connection.
A00.FR.003	If the Charge Point detects that the Central System has accepted a connection with a different profile than the Charge Point is using	The Charge Point SHALL terminate the connection.
A00.FR.004		The security profile SHALL be configured before OCPP communication is enabled.

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.005		Lowering the security profile that is used to a less secure profile is, for security reasons, not part of the OCPP specification, and MUST be done through another method, not via OCPP. OCPP messages SHALL NOT be used for this (e.g. ChangeConfiguration.req or DataTransfer).
A00.FR.006	When a Central System communicates with Charge Points with different security profiles or different versions of OCPP.	The Central System MAY operate the Charge Points via different addresses or ports of the Central System. For instance, the Central System server may have one TCP port for TLS with Basic Authentication, and another port for TLS with Client Side Certificates. In this case there is only one security profile in use per port of the Central System, which is allowed.

NOTE Only securing the OCPP communication is not enough to build a secure Charge Point. All other interfaces to the Charge Point should be equally well secured.

2.3. Unsecured Transport with Basic Authentication Profile - 1

Table 4. Security Profile 1 - Unsecured Transport with Basic Authentication

NO.	TYPE	DESCRIPTION
1	**Name**	Unsecured Transport with Basic Authentication
2	**Profile No.**	1
3	**Description**	The Unsecured Transport with Basic Authentication profile provides a low level of security. Charge Point authentication is done through a username and password. No measures are included to secure the communication channel.
4	**Charge Point Authentication**	For Charge Point authentication HTTP Basic authentication is used.
5	**Central System Authentication**	In this profile, the Central System does not authenticate itself to the Charge Point. The Charge Point has to trust that the server it connects to is indeed the Central System.
6	**Communication Security**	No communication security measures are included in the profile.

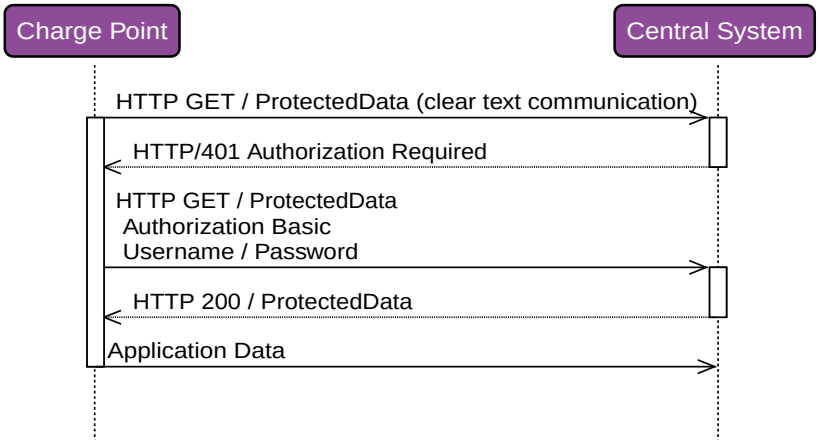

Figure 1. Sequence Diagram: HTTP Basic Authentication sequence diagram

7	**Remark(s)**	The Charge Point should include the same header as used in Basic Auth RFC 2617, while requesting to upgrade the http connection to a websocket connection as described in RFC 6455. The server first needs to validate the Authorization header before upgrading the connection. **Example:** GET /ws HTTP/1.1 Remote-Addr: 127.0.0.1 UPGRADE: websocket CONNECTION: Upgrade HOST: 127.0.0.1:9999 ORIGIN: http://127.0.0.1:9999 SEC-WEBSOCKET-KEY: Pb4obWo2214EfaPQuazMjA== SEC-WEBSOCKET-VERSION: 13 AUTHORIZATION: Basic *<Base64 encoded(<ChargePointId>:<AuthorizationKey>)>*

2.3.1. Unsecured Transport with Basic Authentication Profile - Requirements

Table 5. Security Profile 1 - Unsecured Transport with Basic Authentication - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.201		The Unsecured Transport with Basic Authentication Profile SHOULD only be used in trusted networks.
A00.FR.202		The Charge Point SHALL authenticate itself to the Central System using HTTP Basic authentication [8]
A00.FR.203	A00.FR.202	The client, i.e. the Charge Point, SHALL provide a username and password with every connection request.
A00.FR.204	A00.FR.203	The username SHALL be equal to the Charge Point identity, which is the identifying string of the Charge Point as it uses it in the OCPP-J connection URL.
A00.FR.205	A00.FR.203	The password SHALL be stored in the `AuthorizationKey` Configuration Key. Minimal 16-bytes long, It is strongly advised to be randomly generated binary to get maximal entropy. Hexadecimal represented (20 bytes maximum, represented as a string of up to 40 hexadecimal digits).

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.206	A00.FR.203	With HTTP Basic, the username and password are transmitted in clear text, encoded in base64 only. Hence, it is RECOMMENDED that this mechanism will only be used over connections that are already secured with other means, such as VPNs.

2.4. TLS with Basic Authentication Profile - 2

Table 6. Security Profile 2 - TLS with Basic Authentication

NO.	TYPE	DESCRIPTION
1	**Name**	TLS with Basic Authentication
2	**Profile No.**	2
3	**Description**	In the TLS with Basic Authentication profile, the communication channel is secured using Transport Layer Security (TLS). The Central System authenticates itself using a TLS server certificate. The Charge Points authenticate themselves using HTTP Basic Authentication.
4	**Charge Point Authentication**	For Charge Point authentication HTTP Basic authentication is used. Because TLS is used in this profile, the password will be sent encrypted, reducing the risks of using this authentication method.
5	**Central System Authentication**	The Charge Point authenticates the Central System via the TLS server certificate.
6	**Communication Security**	The communication between Charge Point and Central System is secured using TLS.

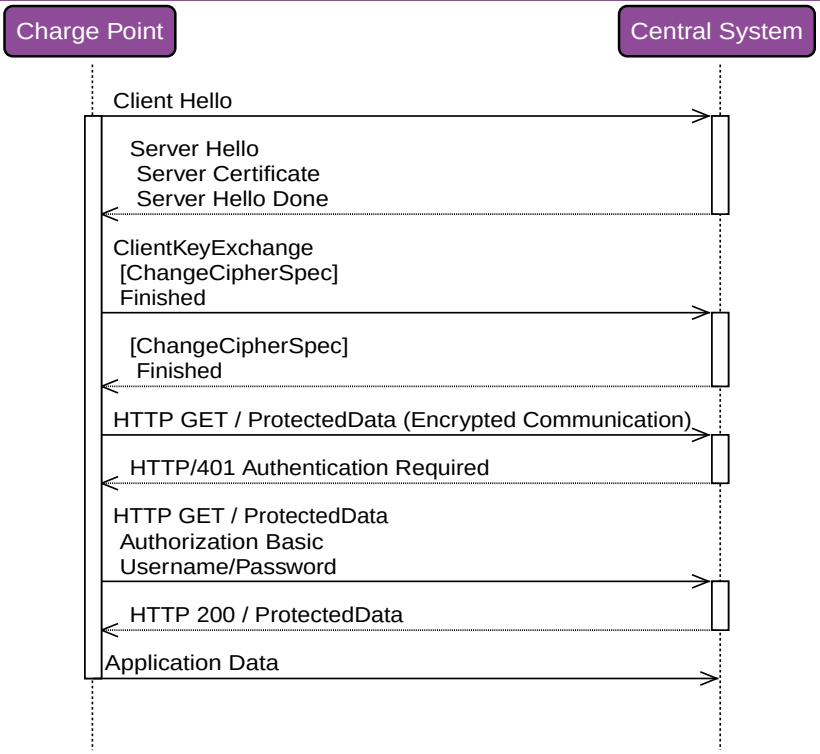

Figure 2. Sequence Diagram: TLS with Basic Authentication sequence diagram

7	**Remark(s)**	TLS allows a number of configurations, not all of which provide sufficient security. The requirements below describe the configurations allowed for OCPP. It is strongly RECOMMENDED to use TLS v1.2 or above for new Charge Points. This also facilitates a later upgrade to OCPP 2.0.1. To provide an adequate level of security for legacy Charge Points that cannot support TLS v1.2 or above, TLS v1.0 or v1.1 MAY be used with cypher suite TLS_RSA_WITH_AES_128_CBC_SHA.

2.4.1. TLS with Basic Authentication Profile - Requirements

Table 7. Security Profile 2 - TLS with Basic Authentication - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.301		The Charge Point SHALL authenticate itself to the Central System using HTTP Basic authentication [8]
A00.FR.302	A00.FR.301	The client, i.e. the Charge Point, SHALL provide a username and password with every connection request.
A00.FR.303	A00.FR.302	The username SHALL be equal to the Charge Point identity, which is the identifying string of the Charge Point as it uses it in the OCPP-J connection URL.
A00.FR.304	A00.FR.302	The password SHALL be stored in the `AuthorizationKey` Configuration Key. Minimal 16-bytes long, It is strongly advised to be randomly generated binary to get maximal entropy. Hexadecimal represented (20 bytes maximum, represented as a string of up to 40 hexadecimal digits).
A00.FR.305		The Central System SHALL act as the TLS server.

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.306		The Central System SHALL authenticate itself by using the Central System certificate as server side certificate.
A00.FR.307		The Charge Point SHALL verify the certification path of the Central System's certificate according to the path validation rules established in Section 6 of [2].
A00.FR.308		The Charge Point SHALL verify that the `commonName` includes the Central System's Fully Qualified Domain Name (FQDN).
A00.FR.309	If the Central System does not own a valid certificate, or if the certification path is invalid	The Charge Point SHALL trigger an InvalidCentralSystemCertificate security event.
A00.FR.310	A00.FR.309	The Charge Point SHALL terminate the connection.
A00.FR.311		The communication channel SHALL be secured using Transport Layer Security (TLS) [3].
A00.FR.312		The Charge Point and Central System SHALL only use TLS v1.2 or above, TLS v1.0/1.1 MAY be used by Charge Points that cannot support TLS v1.2 (NOTE: TLS v1.0/1.1 is not allowed in OCPP 2.0.1).
A00.FR.313		Both of these endpoints SHALL check the version of TLS used.
A00.FR.314	A00.FR.313 AND The Central System detects that the Charge Point only allows connections using an older version of TLS, and TLS v1.0/1.1 not expected for this Charge Point, or only allows SSL	The Central System SHALL terminate the connection.
A00.FR.315	A00.FR.313 AND The Charge Point detects that the Central System only allows connections using an older version of TLS, or only allows SSL	The Charge Point SHALL trigger an InvalidTLSVersion security event AND terminate the connection.
A00.FR.316		TLS SHALL be implemented as in [3] or its successor standards without any modifications.

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.317		The Central System SHALL support at least the following four cipher suites: **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** **TLS_RSA_WITH_AES_128_GCM_SHA256** **TLS_RSA_WITH_AES_256_GCM_SHA384** Note: The Central System will have to provide 2 different certificates to support both Digital Signature Algorithms (RSA and ECDSA). Also when using security profile 3, the Central System should be capable of generating client side certificates for both Digital Signature Algorithms.
A00.FR.318		The Charge Point SHALL support at least the cipher suites: (**TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** AND **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**) OR (**TLS_RSA_WITH_AES_128_GCM_SHA256** AND **TLS_RSA_WITH_AES_256_GCM_SHA384**) OR When the Charge Point supports only TLS v1.0/1.1: **TLS_RSA_WITH_AES_128_CBC_SHA** Note: TLS_RSA does not support forward secrecy, therefore TLS_ECDHE is RECOMMENDED. Furthermore, if the Charge Point detects an algorithm used that is not secure, it SHOULD trigger an InvalidTLSCipherSuite security event (send to the Central System via a SecurityEventNotification.req).
A00.FR.319		The Charge Point and Central System SHALL NOT use cipher suites that use cryptographic primitives marked as unsuitable for legacy use in [1]. This will mean that when one (or more) of the cipher suites described in this specification becomes marked as unsuitable for legacy use, it SHALL NOT be used anymore.
A00.FR.320		The TLS Server and Client SHALL NOT use TLS compression methods to avoid compression side-channel attacks and to ensure interoperability as described in Section 6 of [4].
A00.FR.321	A00.FR.320 AND The Central System detects that the Charge Point only allows connections using one of these suites	The Central System SHALL terminate the connection.
A00.FR.322	A00.FR.320 AND The Charge Point detects that the Central System only allows connections using one of these suites	The Charge Point SHALL trigger an InvalidTLSCipherSuite security event AND terminate the connection.
A00.FR.323	When the Central System terminates the connection because of a security reason	It is RECOMMENDED to log a security event in the Central System.

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.324	When the Central System expects Charge Points with only TLS v1.0/1.1 support	The Central System SHOULD support the cypher suite: **TLS_RSA_WITH_AES_128_CBC_SHA** only for TLS v1.0/1.1 connections.

2.5. TLS with Client Side Certificates Profile - 3

Table 8. Security Profile 3 - TLS with Client Side Certificates

NO.	TYPE	DESCRIPTION
1	**Name**	TLS with Client Side Certificates
2	**Profile No.**	3
3	**Description**	In the TLS with Client Side Certificates profile, the communication channel is secured using Transport Layer Security (TLS). Both the Charge Point and Central System authenticate themselves using certificates.
4	**Charge Point Authentication**	The Central System authenticates the Charge Point via the TLS client certificate.
5	**Central System Authentication**	The Charge Point authenticates the Central System via the TLS server certificate.
6	**Communication Security**	The communication between Charge Point and Central System is secured using TLS.

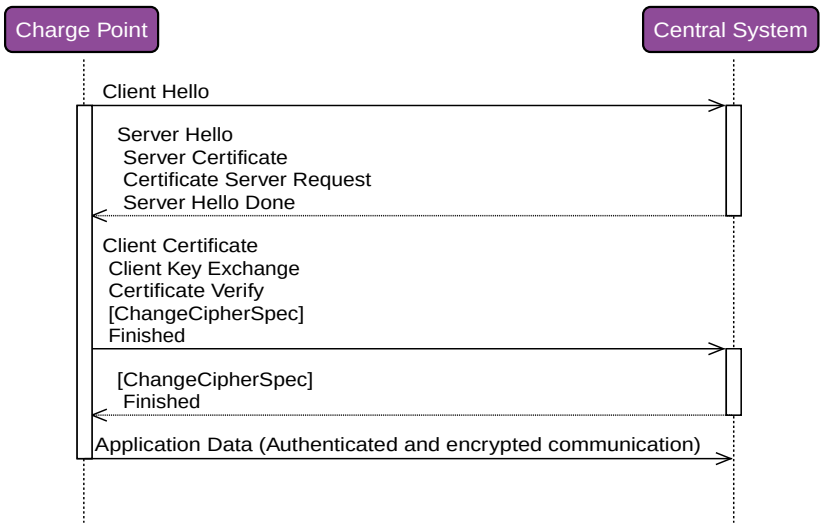

Figure 3. Sequence Diagram: TLS with Client Side Certificates

7	**Remark(s)**	It is strongly RECOMMENDED to use TLS v1.2 or above for new Charge Points. This also facilitates a later upgrade to OCPP 2.0.1. To provide an adequate level of security for legacy Charge Points that cannot support TLS v1.2 or above, TLS v1.0 or v1.1 MAY be used with cypher suite TLS_RSA_WITH_AES_128_CBC_SHA.

2.5.1. TLS with Client Side Certificates Profile - Requirements

Table 9. Security Profile 3 - TLS with Client Side Certificates - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.401		The Charge Point SHALL authenticate itself to the Central System using the Charge Point certificate.
A00.FR.402		The Charge Point certificate SHALL be used as a TLS client side certificate
A00.FR.403		The Central System SHALL verify the certification path of the Charge Point's certificate according to the path validation rules established in Section 6 of [2]
A00.FR.404		The Central System SHALL verify that the certificate is owned by the CPO (or an organization trusted by the CPO) by checking that the O (`organizationName`) RDN in the subject field of the certificate contains the CPO name.
A00.FR.405		The Central System SHALL verify that the certificate belongs to this Charge Point by checking that the CN (`commonName`) RDN in the subject field of the certificate contains the unique Serial Number of the Charge Point
A00.FR.406	If the Charge Point certificate is not owned by the CPO, for instance immediately after installation	it is RECOMMENDED to update the certificate before continuing communication with the Charge Point (also see Installation during manufacturing or installation.)
A00.FR.407	If the Charge Point does not own a valid certificate, or if the certification path is invalid	The Central System SHALL terminate the connection.
A00.FR.408	A00.FR.407	It is RECOMMENDED to log a security event in the Central System.
A00.FR.409		The Central System SHALL act as the TLS server.
A00.FR.410		The Central System SHALL authenticate itself by using the Central System certificate as server side certificate.
A00.FR.411		The Charge Point SHALL verify the certification path of the Central System's certificate according to the path validation rules established in Section 6 of [2].
A00.FR.412		The Charge Point SHALL verify that the `commonName` matches the Central System's Fully Qualified Domain Name (FQDN).
A00.FR.413	If the Central System does not own a valid certificate, or if the certification path is invalid	The Charge Point SHALL trigger an InvalidCentralSystemCertificate security event.
A00.FR.414	A00.FR.413	The Charge Point SHALL terminate the connection.

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.415		The communication channel SHALL be secured using Transport Layer Security (TLS) [3].
A00.FR.416		The Charge Point and Central System SHALL only use TLS v1.2 or above, TLS v1.0/1.1 MAY be used by Charge Points that cannot support TLS v1.2 (NOTE: TLS v1.0/1.1 is not allowed in OCPP 2.0.1).
A00.FR.417		Both of these endpoints SHALL check the version of TLS used.
A00.FR.418	A00.FR.417 AND The Central System detects that the Charge Point only allows connections using an older version of TLS, and TLS v1.0/1.1 not expected for this Charge Point, or only allows SSL	The Central System SHALL terminate the connection.
A00.FR.419	A00.FR.417 AND The Charge Point detects that the Central System only allows connections using an older version of TLS, or only allows SSL	The Charge Point SHALL trigger an InvalidTLSVersion security event AND terminate the connection.
A00.FR.420		TLS SHALL be implemented as in [3] or its successor standards without any modifications.
A00.FR.421		The Central System SHALL support at least the following four cipher suites: **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** **TLS_RSA_WITH_AES_128_GCM_SHA256** **TLS_RSA_WITH_AES_256_GCM_SHA384**
A00.FR.422		The Charge Point SHALL support at least the cipher suites: (**TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** AND **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**) OR (**TLS_RSA_WITH_AES_128_GCM_SHA256** AND **TLS_RSA_WITH_AES_256_GCM_SHA384**) OR When the Charge Point supports only TLS v1.0/1.1: **TLS_RSA_WITH_AES_128_CBC_SHA** Note: TLS_RSA does not support forward secrecy, therefore TLS_ECDHE is preferred. Furthermore, if the Charge Point detects an algorithm used that is not secure, it SHOULD trigger an InvalidTLSCipherSuite security event.

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.423		The Charge Point and Central System SHALL NOT use cipher suites that use cryptographic primitives marked as unsuitable for legacy use in [1]. This will mean that when one (or more) of the cipher suites described in this specification becomes marked as unsuitable for legacy use, it SHALL NOT be used anymore.
A00.FR.424		The TLS Server and Client SHALL NOT use TLS compression methods to avoid compression side-channel attacks and to ensure interoperability as described in Section 6 of [4].
A00.FR.425	A00.FR.424 AND If the Central System detects that the Charge Point only allows connections using one of these suites	The Central System SHALL terminate the connection.
A00.FR.426	A00.FR.424 AND The Charge Point detects that the Central System only allows connections using one of these suites	The Charge Point SHALL trigger an InvalidTLSCipherSuite security event AND terminate the connection.
A00.FR.427		A unique Charge Point certificate SHALL be used for each Charge Point.
A00.FR.428	When the Central System expects Charge Points with only TLS v1.0/1.1 support	The Central System SHOULD support the cypher suite: **TLS_RSA_WITH_AES_128_CBC_SHA** only for TLS v1.0/1.1 connections.
A00.FR.429	When Charge Point supports Security Profile 3	The manufacturer is required to give every Charge Point a unique Serial Number.

2.6. Keys used in OCPP

OCPP uses a number of public private key pairs for its security, see below Table. To manage the keys on the Charge Point, messages have been added to OCPP. Updating keys on the Central System or at the manufacturer is out of scope for OCPP. If TLS with Client Side certificates is used, the Charge Point requires a "Charge Point certificate" for authentication against the Central System.

Table 10. Certificates used in the OCPP security specification

CERTIFICATE	PRIVATE KEY STORED AT	DESCRIPTION
Central System Certificate	Central System	Key used to authenticate the Central System.
Central System Root Certificate	Central System	Certificate used to authenticate the Central System.
Charge Point Certificate	Charge Point	Key used to authenticate the Charge Point.

CERTIFICATE	PRIVATE KEY STORED AT	DESCRIPTION
Firmware Signing Certificate	Manufacturer	Key used to verify the firmware signature.
Manufacturer Root Certificate	Manufacturer	Root certificate for verification of the Manufacturer certificate.

2.6.1. Certificate Properties

Table 11. Certificate Properties requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.501		All certificates SHALL use a private key that provides security equivalent to a symmetric key of at least 112 bits according to Section 5.6.1 of [7]. This is the key size that NIST recommends for the period 2011-2030.
A00.FR.502	A00.FR.501 AND RSA or DSA	This translates into a key that SHALL be at least 2048 bits long.
A00.FR.503	A00.FR.501 AND elliptic curve cryptography	This translates into a key that SHALL be at least 224 bits long.
A00.FR.504		For all cryptographic operations, only the algorithms recommended by BSI in [5], which are suitable for use in future systems, SHALL be used. This restriction includes the signing of certificates in the certificate hierarchy
A00.FR.505		For signing by the certificate authority RSA-PSS, or ECDSA SHOULD be used.
A00.FR.506		For computing hash values the SHA256 algorithm SHOULD be used.
A00.FR.507		The certificates SHALL be stored and transmitted in the X.509 format encoded in Privacy-Enhanced Mail (PEM) format.
A00.FR.508		All certificates SHALL include a serial number.
A00.FR.509		The subject field of the certificate SHALL contain the organization name of the certificate owner in the O (`organizationName`) RDN.
A00.FR.510		For the Central System certificate, the subject field SHALL contain the Fully Qualified Domain Name (FQDN) of the server in the CN (`commonName`) RDN

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.511		For the Charge Point certificate, the subject field SHALL contain a CN (`commonName`) RDN which consists of the unique serial number of the Charge Point. This serial number SHALL NOT be in the format of a URL or an IP address so that Charge Point certificates can be differentiated from Central System certificates. Note: According to RFC 2818 [12], if a `subjectAltName` extension of type `dnsName` is present, that must be used as the identity. This would be incompliant with OCPP. Therefore it SHOULD NOT be used in Charge Point and Central System certificates. It is allowed to use the subjectAltName extension of type dnsName for a Central System, when the Central System has multiple network paths to reached it. (for example, via a private APN + VPN using its IP address in the VPN and via public Internet using a named URL)
A00.FR.512		For all certificates the X.509 Key Usage extension [9] SHOULD be used to restrict the usage of the certificate to the operations for which it will be used.

2.6.2. Certificate Hierarchy

This White Paper adds support for the use of two separate certificate hierarchies:

1. The Charge Point Operator hierarchy which contains the Central System, and Charge Point certificates.
2. The Manufacturer hierarchy which contains the Firmware Signing certificate.

The Central System can update the CPO root certificates stored on the Charge Point using the InstallCertificate.req message.

Table 12. Certificate Hierarchy requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.601		The Charge Point Operator MAY act as a certificate authority for the Charge Point Operator hierarchy
A00.FR.602		The private keys belonging to the CPO root certificates MUST be well protected.
A00.FR.603		As the Manufacturer is usually a separate organization from the Charge Point Operator, a trusted third party SHOULD be used as a certificate authority. This is essential to have non-repudiation of firmware images.

2.6.3. Certificate Revocation

In some cases a certificate may become invalid prior to the expiration of the validity period. Such cases include changes of the organization name, or the compromise or suspected compromise of the certificate's private key. In such cases, the certificate needs to be revoked or indicate it is no longer valid. The revocation of the certificate does not mean that the connection needs to be closed as the the connection can stay open longer than 24 hours.

Different methods are recommended for certificate revocation, see below Table.

Table 13. Recommended revocation methods for the different certificates.

CERTIFICATE	REVOCATION
Central System certificate	Fast expiration
Charge Point certificate	Online verification
Firmware Signing certificate	Online verification

Table 14. Certificate Revocation requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.701		Fast expiration SHOULD be used to revoke the Central System certificate. (See Note 1)
A00.FR.702		The Central System SHOULD use online certificate verification to verify the validity of the Charge Point certificates.
A00.FR.703		It is RECOMMENDED that a separate certificate authority server is used to manage the certificates.
A00.FR.704		The Central System SHALL verify the validity of the certificate with the certificate authority server. (See Note 2)
A00.FR.706		Prior to providing the certificate for firmware validation to the Charge Point, the Central System SHOULD validate both, the certificate and the signed firmware update.

Note 1: With fast expiration, the certificate is only valid for a short period, less than 24 hours. After that the server needs to request a new certificate from the Certificate Authority, which may be the CPO itself (see section Certificate Hierarchy). This prevents the Charge Points from needing to implement revocation lists or online certificate verification. This simplifies the implementation of certificate management at the Charge Point and reduces communication costs at the Charge Point side. By requiring fast expiration, if the certificate is compromised, the impact is reduced to only a short period.

When the certificate chain should becomes compromised, attackers could used forged certificates to trick a Charge Point to connect to a "fake" Central System. By using fast expiration, the time a Charge Point is vulnerable is greatly reduced.

The Charge Point always communicates with the Certificate Authority through the Central System, this way, if the Charge Points is compromised, the Charge Point cannot attack the CA directly.

Note 2: This allows for immediate revocation of Charge Point certificates. Revocation of Charge Point certificates will happen for instance when a Charge Point is removed. This is more common than revoking the Central System certificate, which is normally only done when it is compromised.

Note 3: It is best practice for any certificate authority server to keep track of revoked certificates.

2.6.4. Installation during manufacturing or installation.

Unique credentials should be used to authenticate each Charge Point to the Central System, whether they are the password used for HTTP Basic Authentication (see Charge Point Authentication) or the Charge Point certificate. These unique credentials have to be put on the Charge Point at some point during manufacturing or installation.

Table 15. Certificate Installation requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A00.FR.801		It is RECOMMENDED that the manufacturer initializes the Charge Point with unique credentials during manufacturing.
A00.FR.802	A00.FR.801	The credentials SHOULD be generated using a cryptographic random number generator, and installed in a secure environment.
A00.FR.803	A00.FR.801	The information needed by the CPO to validate the Charge Point credentials SHOULD be sent to the CPO over a secure channel, so that the CPO can import them in the Central System. For example the password. The Certificate Private key is not needed by the CPO and SHOULD NOT be provided to the CPO.
A00.FR.804	If Charge Point certificates are used.	The manufacturer MAY sign these using their own certificate.
A00.FR.805	A00.FR.804	It is RECOMMENDED that the CPO immediately updates the credentials after installation using the methods described in Section A01 - Update Charge Point Password for HTTP Basic Authentication or A02 - Update Charge Point Certificate by request of the Central System.
A00.FR.806	Before the 'factory credentials' have been updated	The Central System MAY restrict the functionality that the Charge Point can use. The Central System can use the BootNotification state: Pending for this. During the Pending state, the Central System can update the credentials.

A01 - Update Charge Point Password for HTTP Basic Authentication

Table 16. A01 - Password Management

NO.	TYPE	DESCRIPTION
1	**Name**	Update Charge Point Password for HTTP Basic Authentication
2	**ID**	A01 *(OCPP 2.0.1)*
3	**Objective(s)**	This use case defines how to use the authorizationKey, the password used to authenticate Charge Points in the Basic and TLS with Basic Authentication security profiles.
4	**Description**	To enable the Central System to configure a new password for HTTP Basic Authentication, the Central System can send a new value for the `AuthorizationKey` Configuration Key.

NO.	TYPE	DESCRIPTION
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System sends a ChangeConfiguration.req(key = AuthorizationKey) to the Charge Point with the `AuthorizationKey` Configuration Key. **2.** The Charge Point responds with ChangeConfiguration.conf and the status *Accepted*. **3.** The Charge Point disconnects it current connection. (Storing any queued messages) **4.** The Charge Point connects to the Central System with the new password.
5	**Prerequisite(s)**	Security Profile: Basic Security Profile or TLS with Basic Authentication in use.
6	**Postcondition(s)**	**Successful postcondition:** The Charge Point has reconnected to the Central System with the new password. **Failure postcondition:** If the Charge Point responds to the ChangeConfiguration.req with a ChangeConfiguration.req with a status other than *Accepted*, the Charge Point will keep using the old credentials. The Central System might treat the Charge Point differently, e.g. by not accepting the Charge Point's boot notifications.

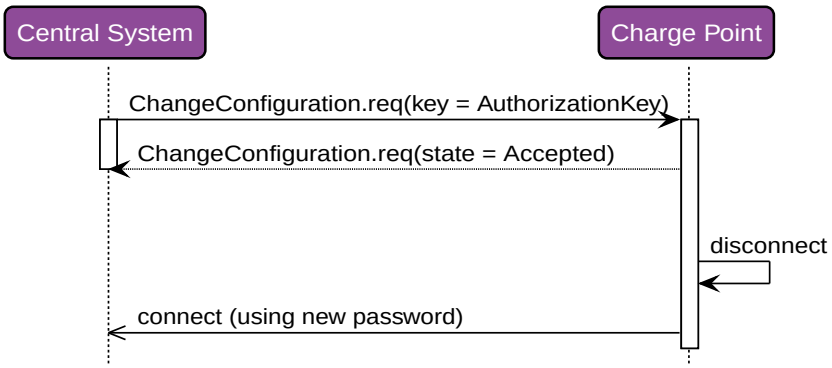

Figure 4. Update Charge Point Password for HTTP Basic Authentication (happy flow)

7	**Error handling**	n/a
8	**Remark(s)**	n/a

A01 - Update Charge Point Password for HTTP Basic Authentication - Requirements

Table 17. A01 - Update Charge Point Password for HTTP Basic Authentication - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A01.FR.01		The Charge Point SHALL store the password in the configuration key `AuthorizationKey`.

ID	PRECONDITION	REQUIREMENT DEFINITION
A01.FR.02		To set a Charge Point's authorization key via OCPP, the Central System SHALL send the Charge Point a ChangeConfiguration.req message with the `AuthorizationKey` Configuration Key.
A01.FR.03	A01.FR.02 AND The Charge Point responds to this ChangeConfiguration.req with a ChangeConfiguration.conf with status *Accepted*.	The Central System SHALL assume that the authorization key change was successful, and no longer accept the credentials previously used by the Charge Point.
A01.FR.04	A01.FR.02 AND The Charge Point responds to this ChangeConfiguration.req with a ChangeConfiguration.conf with status *Rejected* or *NotSupported*.	The Central System SHALL assume that the Charge Point has NOT changed the password. Therefore the Central System SHALL keep accepting the old credentials.
A01.FR.05	A01.FR.04	While the Central System SHALL still accepts a connection from the Charge Point, it MAY restrict the functionality that the Charge Point can use. The Central System can use the BootNotification state: Pending for this. During the Pending state, the Central System can for example retry to update the credentials.
A01.FR.06		Different passwords SHOULD be used for different Charge Points.
A01.FR.07		Passwords SHOULD be generated randomly to ensure that the passwords have sufficient entropy.
A01.FR.08		the Central System SHOULD only store salted password hashes, not the passwords themselves.
A01.FR.09		the Central System SHOULD NOT put the passwords in clear-text in log files or debug information. In this way, if the Central System is compromised not all Charge Point password will be immediately compromised.
A01.FR.10		On the Charge Point the password needs to be stored in clear-text. Extra care SHOULD be taken into storing it securely. Definitions of mechanisms how to securely store the credentials are however not in scope of the OCPP Security Profiles.
A01.FR.11	A01.FR.02	The Charge Point SHALL log the change of `AuthorizationKey` in the Security log.
A01.FR.12	A01.FR.11	The Charge Point SHALL NOT disclose the content of the `AuthorizationKey` in its logging. This is to prevent exposure of key material to persons that may have access to a diagnostics file.

A02 - Update Charge Point Certificate by request of Central System

Table 18. A02 - Update Charge Point Certificate by request of Central System

NO.	TYPE	DESCRIPTION
1	**Name**	Update Charge Point Certificate by request of Central System
2	**ID**	A02 *(OCPP 2.0.1)*
3	**Objective(s)**	To facilitate the management of the Charge Point client side certificate, a certificate update procedure is provided.
4	**Description**	The Central System requests the Charge Point to update its key using ExtendedTriggerMessage.req (SignChargePointCertificate).
	Actors	Charge Point, Central System, Certificate Authority Server
	Scenario description	**1.** The Central System requests the Charge Point to update its certificate using the ExtendedTriggerMessage.req (SignChargePointCertificate) message. **2.** The Charge Point responds with ExtendedTriggerMessage.conf **3.** The Charge Point generates a new public / private key pair. **4.** The Charge Point sends a SignCertificate.req to the Central System. **5.** The Central System responds with SignCertificate.conf, with status *Accepted*. **6.** The Central System forwards the CSR to the Certificate Authority Server. **7.** Certificate Authority Server signs the certificate. **8.** The Certificate Authority Server returns the Signed Certificate to the Central System. **9.** The Central System sends CertificateSigned.req to the Charge Point. **10.** The Charge Point verifies the Signed Certificate. **11.** The Charge Point responds with h to the Central System with the status *Accepted* or *Rejected*.
5	**Prerequisite(s)**	The configuration variable `CpoName` MUST be set.
6	**Postcondition(s)**	**Successful postcondition:** New Client Side certificate installed in the Charge Point. **Failure postcondition:** New Client Side certificate is rejected and discarded.

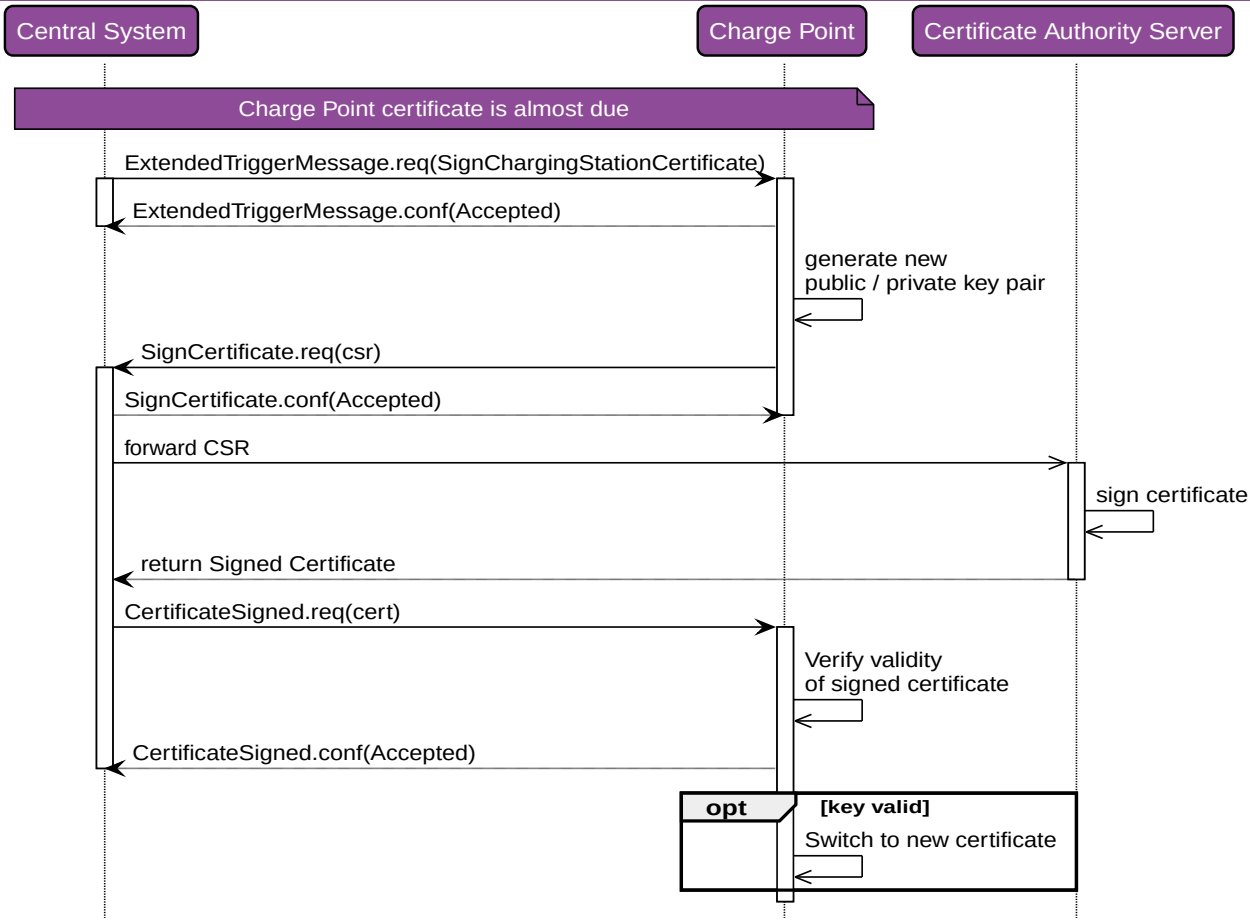

Figure 5. Update Charge Point Certificate

7	**Error handling**	The Central System accepts the CSR request from the Charge Point, before forwarding it to the CA. But when the CA cannot be reached, or rejects the CSR, the Charge Point will never known. The Central System may do some checks on the CSR, but cannot do all the checks that a CA does, and it does not prevent connection timeout to the CA. When something like this goes wrong, either the CA is offline or the CSR send by the Charge Point is not correct, according to the CA. In both cases this is something an operator at the CPO needs to be notified of. The operator then needs to investigate the issue. When resolved, the operator can re-run A02. It is NOT RECOMMENDED to let the Charge Point retry when the certificate is not send within X minutes or hours. When the CSR is incorrect, that will not be resolved automatically. It is possible that only a new firmware will fix this.
8	**Remark(s)**	The CPO may act as a Certification Authority, so the CA Server may be a local server. The applicable Certification Authority SHALL check the information in the CSR. If it is correct, the Certificate Authority SHALL sign the CSR, send it to the CPO, the CPO sends it back to the Charge Point in the CertificateSigned.req message The certificate authority SHOULD implement strong measures to keep the certificate signing private keys secure. Even though the messages CertificateSigned.req (see use cases A02 and A03) and InstallCertificate.req (use case M05 - Install CA Certificate in a Charge Point) are both used to send certificates, their purposes are different. CertificateSigned.req is used to return the the Charge Points *own* public certificate signed by a Certificate Authority. InstallCertificate.req is used to install Root certificates. For (Sub-)CA certificate handling see use cases M03 - Retrieve list of available certificates from a Charge Point, M04 - Delete a specific certificate from a Charge Point, M05 - Install CA certificate in a Charge Point.

A02 - Update Charge Point Certificate by request of Central System - Requirements

Table 19. A02 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A02.FR.01		A key update SHOULD be performed after installation of the Charge Point, to change the key from the one initially provisioned by the manufacturer (possibly a default key).
A02.FR.02	After sending a ExtendedTriggerMessage.conf.	The Charge Point SHALL generate a new public / private key pair using one of the key generation functions described in Section 4.2.1.3 of [6].
A02.FR.03	A02.FR.02	The Charge Point SHALL send the public key in form of a Certificate Signing Request (CSR) as described in RFC 2986 [14] and then PEM encoded, using the SignCertificate.req message.
A02.FR.04		The Central System SHOULD NOT sign the certificate itself, but instead forwards the CSR to a dedicated certificate authority server managing the certificates for the Charge Point infrastructure. The dedicated authority server MAY be operated by the CPO.
A02.FR.05		The private key generated by the Charge Point during the key update process SHALL NOT leave the Charge Point at any time, and SHALL NOT be readable via OCPP or any other (remote) communication connection.
A02.FR.06		The Charge Point SHALL verify the validity of the signed certificate in the CertificateSigned.req message, checking at least the period when the certificate is valid, the properties in Certificate Properties, and that it is part of the Charge Point Operator certificate hierarchy as described in Certificate Hierarchy.
A02.FR.07	If the certificate is not valid.	The Charge Point SHALL discard the certificate, and trigger an InvalidChargePointCertificate security event.
A02.FR.08		The Charge Point SHALL switch to the new certificate as soon as the current date and time is after the 'Not valid before' field in the certificate.
A02.FR.09	If the Charge Point contains more than one valid certificate of the same type.	The Charge Point SHALL use the newest certificate, as measured by the start of the validity period.
A02.FR.10	When the Charge Point has validated that the new certificate works	The Charge Point MAY discard the old certificate. It is RECOMMENDED to store old certificates for one month, as fallback.
A02.FR.11	Upon receipt of a SignCertificate.req AND It is able to process the request	The Central System SHALL set status to *Accepted* in the SignCertificate.conf.

ID	PRECONDITION	REQUIREMENT DEFINITION
A02.FR.12	Upon receipt of a SignCertificate.req AND It is NOT able to process the request	The Central System SHALL set status to *Rejected* in the SignCertificate.conf.
A02.FR.13	A02.FR.03	The Charge Point SHALL put the value of the `CpoName` configuration key in the organizationName (O) RDN in the CSR subject field.

A03 - Update Charge Point Certificate initiated by the Charge Point

Table 20. A03 - Update Charge Point Certificate initiated by the Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Update Charge Point Certificate initiated by the Charge Point
2	**ID**	A03 *(OCPP 2.0.1)*
3	**Objective(s)**	To facilitate the management of the Charge Point client side certificate, a certificate update procedure is provided.
4	**Description**	The Charge Point detects that the 'Charge Point Certificate' it is using will expire in one month. The Charge Point initiates the process to update its key using SignCertificate.req.
	Actors	Charge Point, Central System, Certificate Authority Server
	Scenario description	**1.** The Charge Point detects that the Charge Point certificate is due to expire. **2.** The Charge Point generates a new public / private key pair. **3.** The Charge Point sends a SignCertificate.req to the Central System. **4.** The Central System responds with a SignCertificate.conf, with status *Accepted*. **5.** The Central System forwards the CSR to the Certificate Authority Server. **6.** Certificate Authority Server signs the certificate. **7.** The Certificate Authority Server returns the Signed Certificate to the Central System. **8.** The Central System sends a CertificateSigned.req to the Charge Point. **9.** The Charge Point verifies the Signed Certificate. **10.** The Charge Point responds with a CertificateSigned.conf to the Central System with the status *Accepted* or *Rejected*.
5	**Prerequisite(s)**	The configuration variable `CpoName` MUST be set.
6	**Postcondition(s)**	**Successful postcondition:** New Client Side certificate installed in the Charge Point. **Failure postcondition:** New Client Side certificate is rejected and discarded.

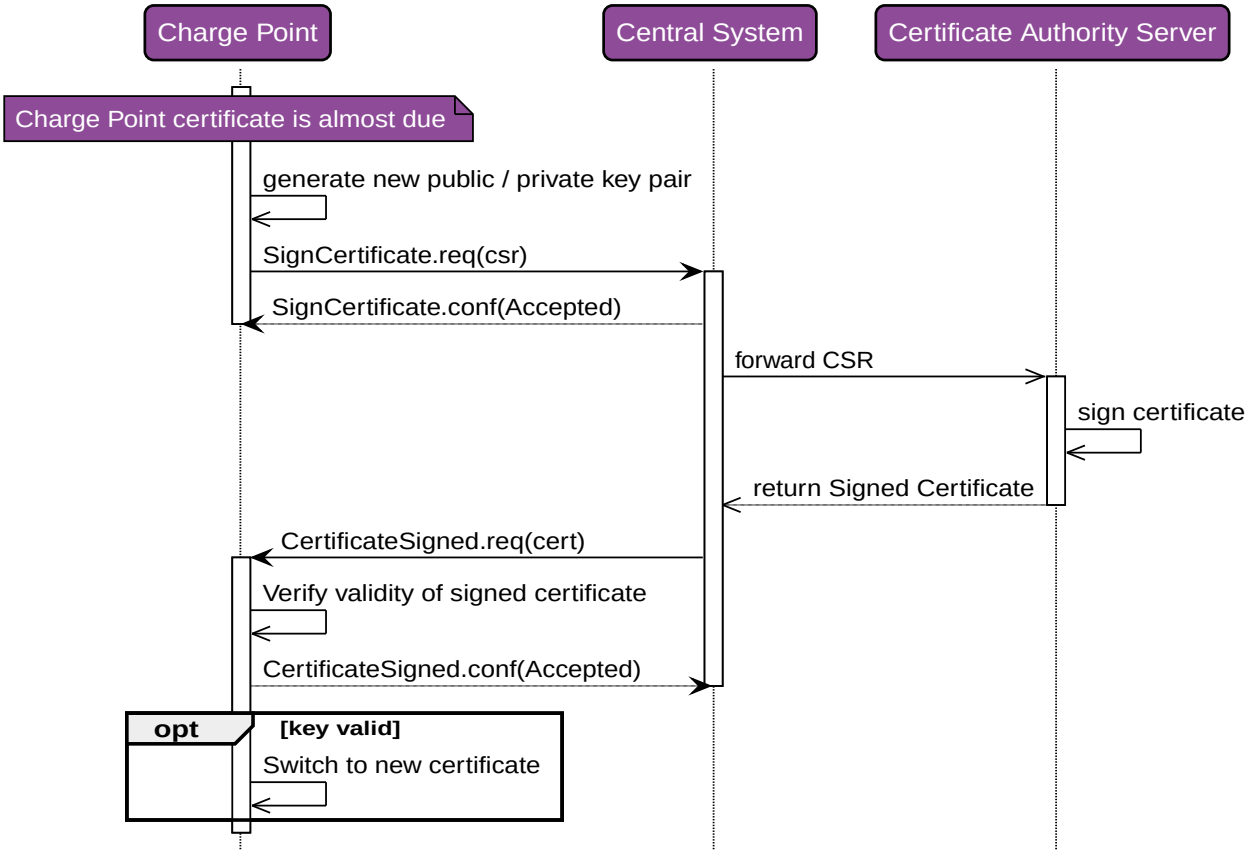

Figure 6. Update Charge Point Certificate initiated by Charge Point

7	**Error handling**	The Central System accepts the CSR request from the Charge Point, before forwarding it to the CA. But when the CA cannot be reached, or rejects the CSR, the Charge Point will never known. The Central System may do some checks on the CSR, but cannot do all the checks that a CA does, and it does not prevent connection timeout to the CA. When something like this goes wrong, either the CA is offline or the CSR send by the Charge Point is not correct, according to the CA. In both cases this is something an operator at the CPO needs to be notified of. The operator then needs to investigate the issue. When resolved, the operator can re-run A02. It is NOT RECOMMENDED to let the Charge Point retry when the certificate is not send within X minutes or hours. When the CSR is incorrect, that will not be resolved automatically. It is possible that only a new firmware will fix this.
8	**Remark(s)**	Same remarks as in A02 - Update Charge Point Certificate by request of Central System apply.

A03 - Update Charge Point Certificate initiated by the Charge Point - Requirements

Table 21. A03 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A03.FR.01		A key update MAY be performed after installation of the Charge Point, to change the key from the one initially provisioned by the manufacturer (possibly a default key).
A03.FR.02	When the Charge Point detects that the current Charge Point certificate will expire in one month.	The Charge Point SHALL generate a new public / private key pair using one of the key generation functions described in Section 4.2.1.3 of [6].

ID	PRECONDITION	REQUIREMENT DEFINITION
A03.FR.03	A03.FR.02	The Charge Point SHALL send the public key in form of a Certificate Signing Request (CSR) as described in RFC 2986 [14] and then PEM encoded, using the SignCertificate.req message.
A03.FR.04		The Central System SHOULD NOT sign the certificate itself, but instead forwards the CSR to a dedicated certificate authority server managing the certificates for the Charge Point infrastructure. The dedicated authority server MAY be operated by the CPO.
A03.FR.05		The private key generated by the Charge Point during the key update process SHALL NOT leave the Charge Point at any time, and SHALL NOT be readable via OCPP or any other (remote) communication connection.
A03.FR.06		The Charge Point SHALL verify the validity of the signed certificate in the CertificateSigned.req message, checking at least the period when the certificate is valid, the properties in Certificate Properties, and that it is part of the Charge Point Operator certificate hierarchy as described in Certificate Hierarchy.
A03.FR.07	If the certificate is not valid.	The Charge Point SHALL discard the certificate, and trigger an InvalidChargePointCertificate security event.
A03.FR.08		The Charge Point SHALL switch to the new certificate as soon as the current date and time is after the 'Not valid before' field in the certificate.
A03.FR.09	If the Charge Point contains more than one valid certificate of the same type.	The Charge Point SHALL use the newest certificate, as measured by the start of the validity period.
A03.FR.10	When the Charge Point has validated that the new certificate works	The Charge Point MAY discard the old certificate. It is RECOMMENDED to store old certificates for one month, as fallback.
A03.FR.11	Upon receipt of a SignCertificate.req AND It is able to process the request	The Central System SHALL set status to *Accepted* in the SignCertificate.conf.
A03.FR.12	Upon receipt of a SignCertificate.req AND It is NOT able to process the request	The Central System SHALL set status to *Rejected* in the SignCertificate.conf.
A03.FR.13	A03.FR.03	The Charge Point SHALL put the value of `CpoName` in the organizationName RDN in the CSR subject field.

A05 - Upgrade Charge Point Security Profile

Table 22. A05 - Upgrade Charge Point Security Profile

NO.	TYPE	DESCRIPTION
1	**Name**	Upgrade Charge Point Security Profile
2	**ID**	A05 *(OCPP 2.0.1)*
3	**Objective(s)**	Upgrade the security profile used by a Charge Point to a higher profile.
4	**Description**	The CPO wants to increase the security of the OCPP connection between Central System and a Charge Point. This use case is especially relevant when migrating from OCPP 1.6 without security profiles to OCPP 1.6 with security profiles, before migrating to a security profile the prerequisites, like installed certificates or password need to be configured. The CPO ensures the prerequisite(s) for going to a higher security certificates are met before sending the command to change to a higher security profile. the Charge Point reconnects to the Central System using the higher security profile.
	Actors	Charge Point, Central System, CPO
	Scenario description	**1.** CPO command the Central System to upgrade a Charge Point to a higher Security Profile. **2.** The Central System sends a ChangeConfiguration.req for configuration key: `SecurityProfile` with a new (higher) value to the Charge Point. **3.** The Charge Point checks all the prerequisites for the new Security Profile. **4.** The Charge Point responds with ChangeConfiguration.conf. **5.** The Charge Point disconnects it's current connection the Central System. **6.** The Charge Point connects to the Central System using the new Security Profile.
5	**Prerequisite(s)**	Configuration Key: `SecurityProfile` available.
6	**Postcondition(s)**	**Successful postcondition:** The Charge Point is using the higher security profile. **Failure postcondition:** The Charge Point is NOT using the higher security profile.

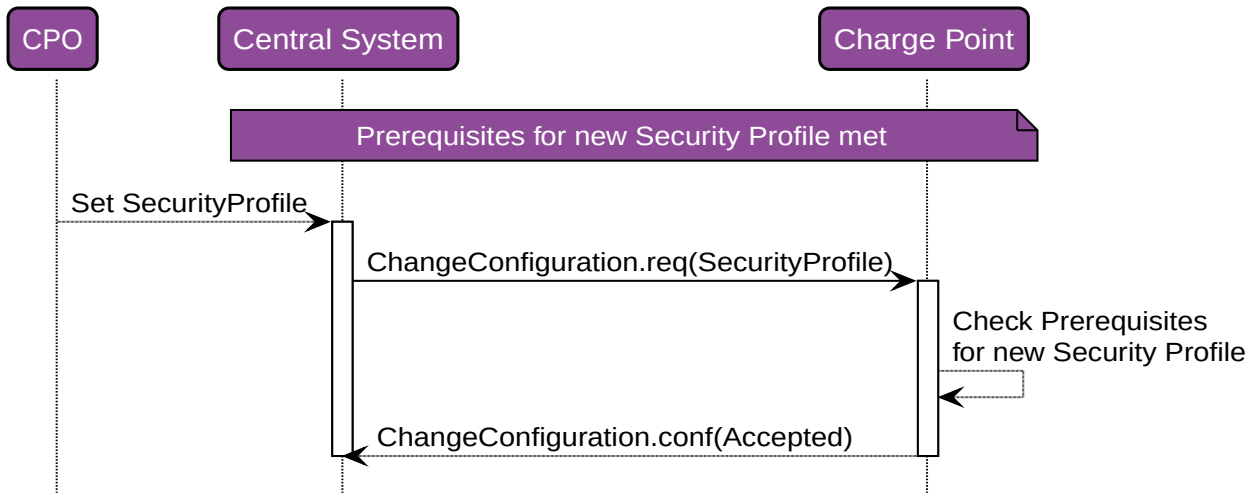

Figure 7. Upgrade Charge Point Certificate initiated by Charge Point

7	**Error handling**	If the Charge Point is unable to connect to the Central System using the configured (higher) security profile, it SHOULD fallback to its previous security profile settings. This is to prevent that the Charge Point will become unable to reconnect to the Central System on its own.
8	**Remark(s)**	For security reasons it is not allowed to change to a lower Security Profile over OCPP.

A05 - Upgrade Charge Point Security Profile - Requirements

Table 23. A05 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
A05.FR.01	Charge Point receives ChangeConfiguration.req for `SecurityProfile` with a value lower or equal to the current value.	The Charge Point SHALL respond with ChangeConfiguration.conf(Rejected), and not change the value for `SecurityProfile` and/or reconnect to the Central System.
A05.FR.02	Charge Point receives ChangeConfiguration.req for `SecurityProfile` with a value higher then the current value AND new value is 1 or 2 AND configuration key: `AuthorizationKey` does not contain a value (that meets the requirements for `AuthorizationKey`)	The Charge Point SHALL respond with ChangeConfiguration.conf(Rejected), and not change the value for `SecurityProfile` and/or reconnect to the Central System.
A05.FR.03	Charge Point receives ChangeConfiguration.req for `SecurityProfile` with a value higher then the current value AND new value is 2 or 3 AND No valid CentralSystemRootCertificate installed	The Charge Point SHALL respond with ChangeConfiguration.conf(Rejected), and not change the value for `SecurityProfile` and/or reconnect to the Central System.
A05.FR.04	Charge Point receives ChangeConfiguration.req for `SecurityProfile` with a value higher then the current value AND new value is 3 AND No valid ChargePointCertificate installed	The Charge Point SHALL respond with ChangeConfiguration.conf(Rejected), and not change the value for `SecurityProfile` and/or reconnect to the Central System.
A05.FR.05	Charge Point receives ChangeConfiguration.req for `SecurityProfile` with a value higher then the current value AND all prerequisites are met	The Charge Point SHALL respond with ChangeConfiguration.conf(Accepted)
A05.FR.06	A05.FR.05	The Charge Point SHALL disconnect from the Central System
A05.FR.07	A05.FR.06	The Charge Point SHALL reconnect the Central System with the new Security Profile
A05.FR.08	A05.FR.07 AND The Charge Point was unable to connect to the Central System	The Charge Point SHOULD fallback to its previous security profile setting.

ID	PRECONDITION	REQUIREMENT DEFINITION
A05.FR.09	A05.FR.07 AND The Charge Point was able to successfully connect to the Central System	The Central System SHALL NOT allow the Charge Point to connect with a lower security profile anymore.

M03 - Retrieve list of available certificates from a Charge Point

Table 24. M03 - Retrieve list of available certificates from a Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Retrieve list of available certificates from a Charge Point
2	**ID**	M03 *(OCPP 2.0.1)*
3	**Objective(s)**	To enable the Central System to retrieve a list of available certificates from a Charge Point.
4	**Description**	To facilitate the management of the Charge Point's installed certificates, a method of retrieving the installed certificates is provided. The Central System requests the Charge Point to send a list of installed certificates
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System requests the Charge Point to send a list of installed certificates by sending a GetInstalledCertificateIds.req **2.** The Charge Point responds with a GetInstalledCertificateIds.conf
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The Central System received a list of installed certificates

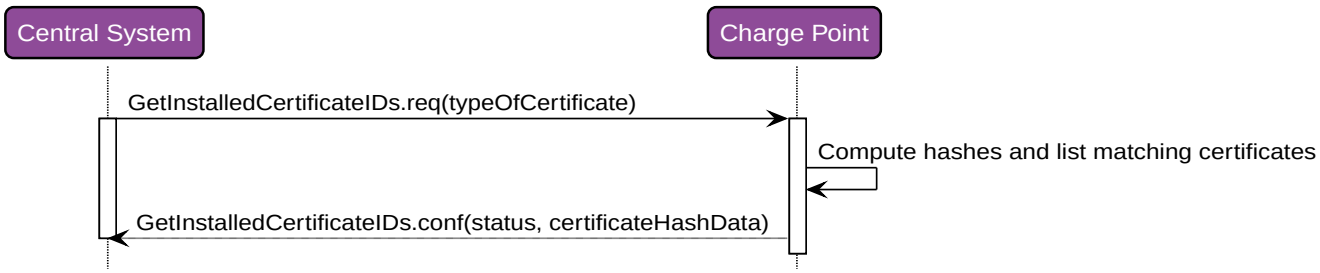

Figure 8. Retrieve list of available certificates from a Charge Point

7	**Error handling**	n/a
8	**Remark(s)**	For installing the Charge Point Certificate, see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point.

M03 - Retrieve list of available certificates from a Charge Point - Requirements

Table 25. M03 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
M03.FR.01	After receiving a GetInstalledCertificateIds.req	The Charge Point SHALL respond with a GetInstalledCertificateIds.conf.
M03.FR.02	M03.FR.01 AND No certificate matching *certificateType* was found	The Charge Point SHALL indicate this by setting *status* in the GetInstalledCertificateIds.conf to *NotFound*.
M03.FR.03	M03.FR.01 AND A certificate matching *certificateType* was found	The Charge Point SHALL indicate this by setting *status* in the GetInstalledCertificateIds.conf to *Accepted*.
M03.FR.04	M03.FR.03	The Charge Point SHALL include the hash data for each matching installed certificate in the GetInstalledCertificateIds.conf.

M04 - Delete a specific certificate from a Charge Point

Table 26. M04 - Delete a specific certificate from a Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Delete a specific certificate from a Charge Point
2	**ID**	M04 *(OCPP 2.0.1)*
3	**Objective(s)**	To enable the Central System to request the Charge Point to delete an installed certificate.
4	**Description**	To facilitate the management of the Charge Point's installed certificates, a method of deleting an installed certificate is provided. The Central System requests the Charge Point to delete a specific certificate.
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System requests the Charge Point to delete an installed certificate by sending a DeleteCertificate.req. **2.** The Charge Point responds with a DeleteCertificate.conf.
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The requested certificate was deleted from the Charge Point.

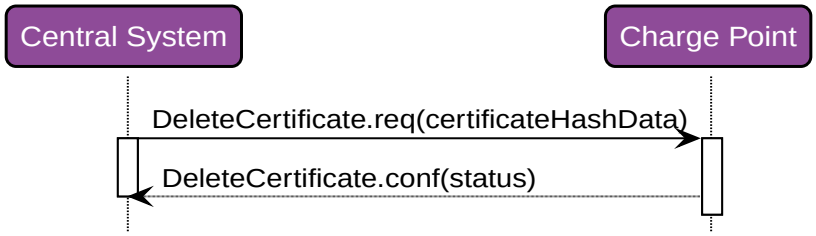

Figure 9. Delete Installed Certificate

7	**Error handling**	n/a
8	**Remark(s)**	For installing the Charge Point Certificate, see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point. It is possible to delete the last (every) installed CentralSystemRootCertificates, when all CentralSystemRootCertificates are deleted, the Charge Point cannot validate Central System Certificates, so it will not be able to connect to a Central System. Before a Central System would ever send a DeleteCertificate.req that would delete the last/all CentralSystemRootCertificates the Central System is ADVICED to make very sure that this is what is really wanted. It is possible to delete the last (every) installed ManufacturerRootCertificates, when all ManufacturerRootCertificates are deleted, no "Signed Firmware" can be installed in the Charge Point.

M04 - Delete a specific certificate from a Charge Point - Requirements

Table 27. M04 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
M04.FR.01	After receiving a DeleteCertificate.req	The Charge Point SHALL respond with a DeleteCertificate.conf.
M04.FR.02	M04.FR.01 AND The requested certificate was found	The Charge Point SHALL delete it, and indicate success by setting 'status' to 'Success' in the DeleteCertificate.conf.
M04.FR.03	M04.FR.01 AND The deletion fails	The Charge Point SHALL indicate failure by setting 'status' to 'Failed' in the DeleteCertificate.conf.
M04.FR.04	M04.FR.01 AND The requested certificate was not found	The Charge Point SHALL indicate failure by setting 'status' to 'NotFound' in the DeleteCertificate.conf.
M04.FR.05		Deletion of the *Charge Point Certificate* SHALL NOT be possible via a DeleteCertificate.req.
M04.FR.06	M04.FR.01 AND Certificate to delete is a CentralSystemRootCertificate AND This CentralSystemRootCertificate is currently in use for validation of the connection the the Central System	The Charge Point SHALL reject the request by setting 'status' to 'Failed' in the DeleteCertificate.conf.

ID	PRECONDITION	REQUIREMENT DEFINITION
M04.FR.07	When deleting a certificate	The Central System SHALL use the *hashAlgorithm*, which was used to install the certificate.

M05 - Install CA certificate in a Charge Point

Table 28. M05 - Install CA certificate in a Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Install CA certificate in a Charge Point
2	**ID**	M05 *(OCPP 2.0.1)*
3	**Objective(s)**	To facilitate the management of the Charge Point's installed certificates, a method to install a new CA certificate.
4	**Description**	The Central System requests the Charge Point to install a new Central System root certificate or Manufacturer root certificate.
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System requests the Charge Point to install a new certificate by sending an InstallCertificate.req. **2.** The Charge Point responds with an InstallCertificate.conf.
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The new certificate was installed in the Charge Point trust store.

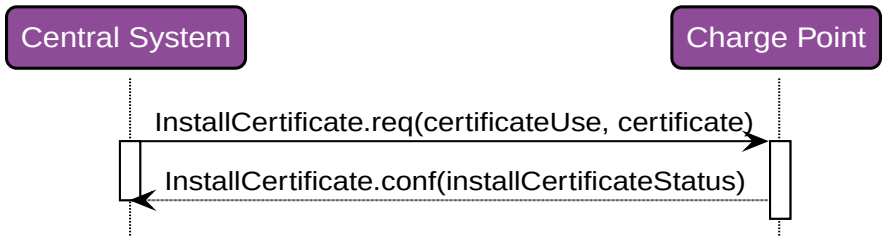

Figure 10. Install CA certificate in a Charge Point

7	**Error handling**	n/a

8	**Remark(s)**	Even though the messages CertificateSigned.req (see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point) and InstallCertificate.req (use case M05) are both used to send certificates, their purposes are different. CertificateSigned.req is used to return the the Charge Points *own* public certificate signed by a Certificate Authority. InstallCertificate.req is used to install Root certificates. For installing the Charge Point Certificate, see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point. It is allowed to have multiple certificates of the same type installed.

M05 - Install CA certificate in a Charge Point - Requirements

Table 29. M05 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
M05.FR.01	After receiving an InstallCertificate.req	The Charge Point SHALL attempt to install the certificate and respond with an InstallCertificate.conf.
M05.FR.02	M05.FR.01 AND The installation was successful	The Charge Point SHALL indicate success by setting 'status' to 'Accepted' in the InstallCertificate.conf.
M05.FR.03	M05.FR.01 AND Current amount of install certificates >= `CertificateStoreMaxLength`	The Charge Point SHALL indicate failure (no more space to install more certificates) by setting 'status' to 'Rejected' in the InstallCertificate.conf
M05.FR.04	M05.FR.01 AND The installation failed	The Charge Point SHALL indicate failure by setting 'status' to 'Failed' in the InstallCertificate.conf.
M05.FR.06	M05.FR.01 AND The certificate is invalid and/or incorrect.	The Charge Point SHALL indicate rejection by setting 'status' to 'Rejected' in the InstallCertificate.conf.
M05.FR.08	When `AdditionalRootCertificateCheck` is true	Only one certificate (plus a temporarily fallback certificate) of certificateType CentralSystemRootCertificate is allowed to be installed at a time.
M05.FR.09	When `AdditionalRootCertificateCheck` is true AND installing a new certificate of certificateType CentralSystemRootCertificate	The new Central System Root certificate SHALL replace the old Central System Root certificate AND the new Root Certificate MUST be signed by the old Root Certificate it is replacing
M05.FR.10	M05.FR.09 AND the new Central System Root certificate is NOT signed by the old Central System Root certificate	The Charge Point SHALL NOT install the new Central System Root Certificate and respond with status *Rejected*.

ID	PRECONDITION	REQUIREMENT DEFINITION
M05.FR.11	M05.FR.09 AND the new Central System Root certificate is signed by the old Central System Root certificate	The Charge Point SHALL install the new Central System Root Certificate AND temporarily keep the old Central System Root certificate as a fallback certificate AND respond with status *Accepted*
M05.FR.12	M05.FR.11 AND the Charge Point successfully connected to the Central System using the new Central System Root certificate	The Charge Point SHALL remove the old Central System Root (fallback) certificate.
M05.FR.13	M05.FR.11 AND The Charge Point is attempting to reconnect to the Central System, but determines that the server certificate provided by the Central System is invalid when using the new Central System Root certificate to verify it	The Charge Point SHALL try to use the old Central System Root (fallback) certificate to verify the server certificate.

3. Security events/logging

A04 - Security Event Notification

Table 30. A04 - Security Event Notification

NO.	TYPE	DESCRIPTION
1	**Name**	Security Event Notification
2	**ID**	A04 *(OCPP 2.0.1)*
3	**Objective(s)**	To inform the Central System of critical security events.
4	**Description**	This use case allows the Charge Point to immediately inform the Central System of changes in the system security.
	Actors	Central System, Charge Point
	Scenario description	**1.** A *critical* security event happens. **2.** The Charge Point sends a SecurityEventNotification.req to the Central System. **3.** The Central System responds with SecurityEventNotification.conf to the Charge Point.
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The Charge Point *successfully* informed the Central System of critical security events by sending a SecurityEventNotification.req to the Central System.

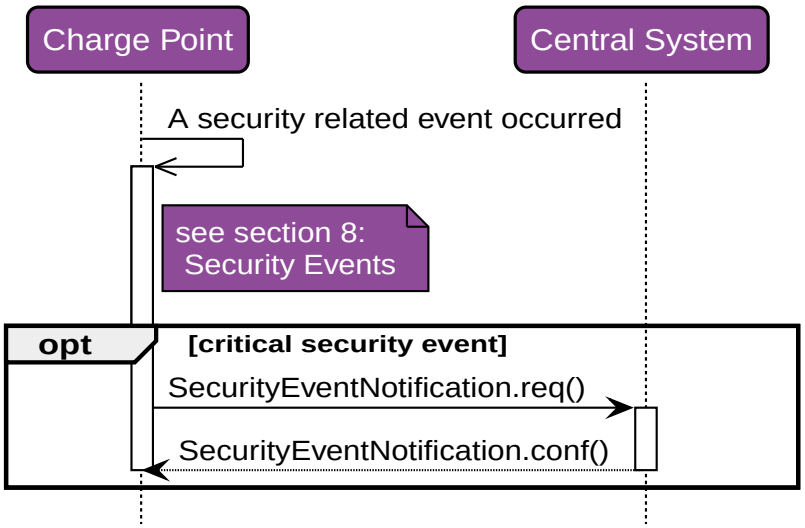

Figure 11. Security Event Notification

7	**Error handling**	n/a

8	**Remark(s)**	A list of security related events and their 'criticality' is provided at Security Events

A04 - Security Event Notification - Requirements

Table 31. A04 - Security Event Notification - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
A04.FR.01	When a *critical* security event happens	The Charge Point SHALL inform the Central System of the security events by sending a SecurityEventNotification.req, to the Central System.	
A04.FR.02	A04.FR.01 AND the Charge Point is disconnected.	Security event notifications MUST be queued with a guaranteed delivery at the Central System.	
A04.FR.03	A04.FR.01	The Central System SHALL confirm the receipt of the notification using the SecurityEventNotification.conf message.	
A04.FR.04	When a security event happens (also none-critical)	The Charge Point SHALL store the security event in a security log.	It is recommended to implement this in a rolling format.

N01 - Retrieve Log Information

Table 32. N01 - Retrieve Log Information

NO.	TYPE	DESCRIPTION
1	**Name**	Retrieve Log
2	**ID**	N01 *(OCPP 2.0.1)*
3	**Objective(s)**	To enable the Central System retrieving of log information from a Charge Point.
4	**Description**	This use case covers the functionality of getting log information from a Charge Point. The Central System can request a Charge Point to upload a file with log information to a given location (URL). The format of this log file is not prescribed. The Charge Point uploads a log file and gives information about the status of the upload by sending status notifications to the Central System.
	Actors	Charge Point, Central System

NO.	TYPE	DESCRIPTION
	Scenario description	**1.** The Central System sends a GetLog.req to the Charge Point. **2.** The Charge Point responds with a GetLog.conf. **3.** The Charge Point sends a LogStatusNotification.req with the status Uploading **4.** The Central System responds with a LogStatusNotification.conf acknowledging the status update request. **5.** Uploading of the diagnostics files. **6.** The Charge Point sends LogStatusNotification.req with the status *Uploaded*. **7.** The Central System responds with LogStatusNotification.conf, acknowledging the status update request. **8.** The Charge Point returns to *Idle* status.
5	**Prerequisite(s)**	- Requested information (either DiagnosticsLog or SecurityLog) is available for upload. - URL to upload file to is reachable and exists.
6	**Postcondition(s)**	**Successful postcondition:** Log file *Successfully* uploaded. **Failure postcondition:** Log file *not Successfully* uploaded and *Failed*.

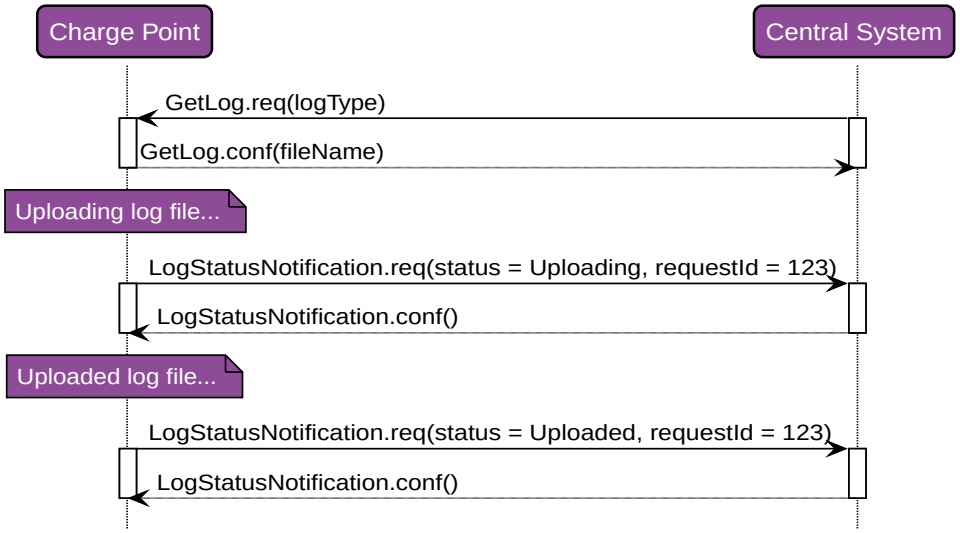

Figure 12. Sequence Diagram: Get Security Log

7	**Error handling**	When the upload fails, and the transfer protocol supports "resume", it is recommended that the Charge Point tries "resume" before aborting the upload.

8	**Remark(s)**	When a Charge Point is requested to upload a log file, the Central System supplies in the request an URL where the Charge Point SHALL upload the file. The URL also contains the protocol which must be used to upload the file. It is recommended that the log file is uploaded via FTP or FTPS. FTP(S) is better optimized for large binary data than HTTP. Also FTP(S) has the ability to resume uploads. In case an upload is interrupted, the Charge Point can resume uploading after the part it already has uploaded. The FTP URL is of format: *ftp://User:password@host:port /path* in which the parts *User*:*password*@, :*password* or :*port* may be excluded. The Charge Point has an optional Configuration Key that reports which file transfer protocols it supports: `SupportedFileTransferProtocols`. The format of the log file is not prescribed. FTP needs to be able to use Passive FTP, to be able to transverse over as much different typologies as possible.

N01 - Retrieve Log Information - Requirements

Table 33. N01 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
N01.FR.01	Upon receipt of a GetLog.req AND if the requested log information is available	The Charge Point SHALL respond with a GetLog.conf stating the name of the file and status *Accepted*.	
N01.FR.02	N01.FR.01	The Charge Point SHALL start uploading a single log file to the specified location	
N01.FR.03	N01.FR.02 AND The GetLog.req contained logType *SecurityLog*	The Charge Point SHALL upload its security log	
N01.FR.04	N01.FR.02 AND The GetLog.req contained logType *DiagnosticsLog*	The Charge Point SHALL upload its diagnostics.	
N01.FR.05	When a security event happens	The Charge Point SHALL log this event in its security log. See Section 8. Security Events for a list of security events.	
N01.FR.07		Every LogStatusNotification.req that is sent for the upload of a specific log SHALL contain the same requestId as the GetLog.req that started this log upload.	
N01.FR.08	When uploading a log document is started	The Charge Point SHALL send a LogStatusNotification.req with status *Uploading*.	
N01.FR.09	When a log document is uploaded successfully	The Charge Point SHALL send a LogStatusNotification.req with status *Uploaded*.	

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
N01.FR.10	When uploading a log document failed	The Charge Point SHALL send a LogStatusNotification.req with status *UploadFailed*, *BadMessage*, *PermissionDenied* OR *NotSupportedOperation*.	It is RECOMMENDED to send a status that describes the reason of failure as precise as possible.
N01.FR.11	When a Charge Point is uploading a log file AND the Charge Point receives a new GetLog.req	The Charge Point SHOULD cancel the ongoing log file upload AND respond with status *AcceptedCanceled*.	
N01.FR.12		The field *requestId* in LogStatusNotification.req is mandatory, unless the message was triggered by an ExtendedTriggerMessage.req AND there is no log upload ongoing.	

4. Secure firmware update

L01 - Secure Firmware Update

Table 34. L01 - Secure Firmware Update

NO.	TYPE	DESCRIPTION
1	**Name**	Secure Firmware Update
2	**ID**	L01
3	**Objective(s)**	Download and install a Secure firmware update.
4	**Description**	Illustrate how a Charge Point processes a Secure firmware update.
	Actors	Central System, Charge Point
	Scenario description	**1.** The Central System sends a SignedUpdateFirmware.req message that contains the location of the firmware, the time after which it should be retrieved, and information on how many times the Charge Point should retry downloading the firmware. **2.** The Charge Point verifies the validity of the certificate against the Manufacturer root certificate. **3.** If the certificate is not valid or could not be verified, the Charge Point aborts the firmware update process and sends a SignedUpdateFirmware.conf with status InvalidCertificate (or status RevokedCertificate when the certificate has been revoked) and a SecurityEventNotification.req with the security event InvalidFirmwareSigningCertificate. If the certificate is valid, the Charge Point starts downloading the firmware, and sends a SignedFirmwareStatusNotification.req with status Downloading. **4.** If the Firmware successfully downloaded, the Charge Point sends a SignedFirmwareStatusNotification.req with status Downloaded. Otherwise, it sends a SignedFirmwareStatusNotification.req with status DownloadFailed. **5.** If the verification is successful, the Charge Point sends a SignedFirmwareStatusNotification.req with status Installing. If the verification of the firmware fails or if a signature is missing entirely, the Charge Point sends a SignedFirmwareStatusNotification.req with status InvalidSignature and a SecurityEventNotification.req with the security event InvalidFirmwareSignature. **6.** If the installation is successful, the Charge Point sends a SignedFirmwareStatusNotification.req with status Installed. Otherwise, it sends a SignedFirmwareStatusNotification.req with status InstallationFailed.
5	**Prerequisite(s)**	The Charge Point Manufacturer provided a firmware update, signing certificate and signature.
6	**Postcondition(s)**	**Successful postcondition:** The firmware is updated and the Charge Point is in *Installed* status. **Failure postconditions:** The certificate is not valid or could not be verified and the Charge Point is in *InvalidCertificate* status. Downloading the firmware failed and the Charge Point is in *DownloadFailed* status. The verification of the firmware's digital signature failed and the Charge Point is in *InvalidSignature* status. The installation of the firmware is not successful and the Charge Point is in *InstallationFailed* status.

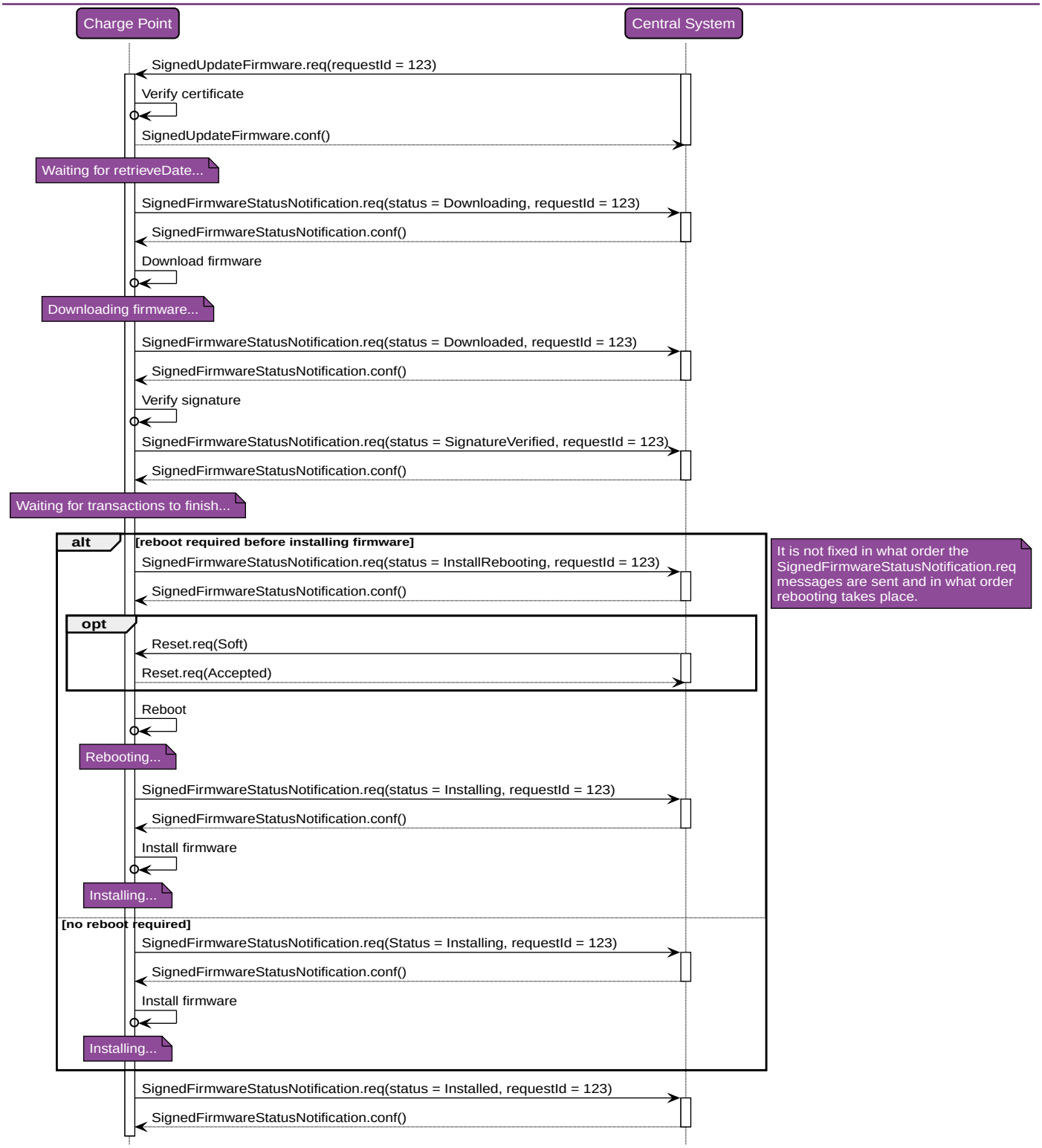

Figure 13. Sequence diagram secure firmware upgrade (happy flow)

7	**Error handling**	n/a

8	**Remark(s)**	Measures SHOULD be taken to secure the firmware when it is stored on a server or workstation. The Charge Point has a required Configuration Key that reports which file transfer protocols it supports: `SupportedFileTransferProtocols` The requirements for the Firmware Signing Certificate are described in the: Certificate Properties section. The manufacturer SHALL NOT use intermediate certificates for the firmware signing certificate in the Charge Point. FTP needs to be able to use Passive FTP, to be able to transverse over as much different typologies as possible.

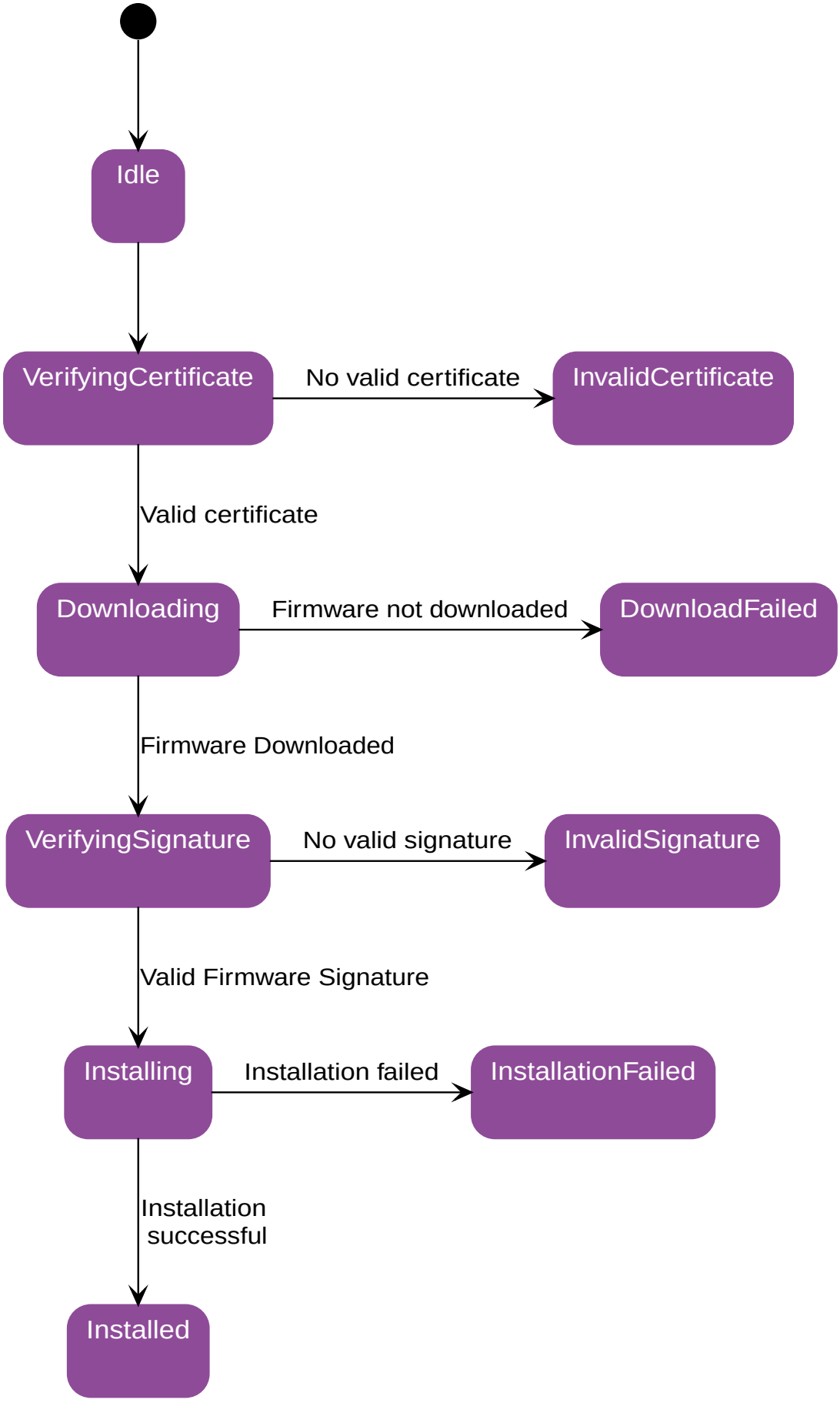

Figure 14. Firmware update process

L01 - Secure Firmware Update - Requirements

Table 35. L01 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.01	Whenever the Charge Point enters a new state in the firmware update process.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req message to the Central System with this new status. What reason to use is described in the description of FirmwareStatusEnumType.	
L01.FR.02	When the Charge Point enters the Invalid Certificate state in the firmware process.	The Charge Point SHALL send a SecurityEventNotification.req message to the Central System with the security event InvalidFirmwareSigningCertificate.	
L01.FR.03	When the Charge Point enters the Invalid Signature state.	The Charge Point SHALL send a SecurityEventNotification.req message to the Central System with the security event InvalidFirmwareSignature.	
L01.FR.04	When the Charge Point has successfully downloaded the new firmware	The signature SHALL be validated, by calculating the signature over the entire firmware file using the RSA-PSS or ECSchnorr algorithm for signing, and the SHA256 algorithm for calculating hash values.	
L01.FR.05	L01.FR.04 AND installDateTime is not set	The Charge Point SHALL install the new firmware as soon as it is able to.	
L01.FR.06	L01.FR.05 AND The Charge Point has ongoing transactions AND When it is not possible to continue charging during installation of firmware	The Charge Point SHALL wait until all transactions have ended, before commencing installation.	
L01.FR.07	L01.FR.06	The Charge Point SHALL set all connectors that are not in use to UNAVAILABLE while the Charge Point waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.	
L01.FR.08		It is RECOMMENDED that the firmware is sent encrypted to the Charge Point. This can either be done by using a secure protocol (such as HTTPS, SFTP, or FTPS) to send the firmware, or by encrypting the firmware itself before sending it.	

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.09		Firmware updates SHALL be digitally protected to ensure authenticity and to provide proof of origin.	This protection is achieved by applying a digital signature over the hash value of the firmware image. Ideally, this signature is already computed by the manufacturer. This way proof of origin of the firmware image can be tracked back to the original author of the firmware.
L01.FR.10		Every SignedFirmwareStatusNotification.req that is sent for a specific firmware update SHALL contain the same requestId as the SignedUpdateFirmware.req that started this firmware update.	
L01.FR.11		For security purposes the Central System SHALL include the Firmware Signing certificate (see Keys used in OCPP) in the SignedUpdateFirmware.req.	
L01.FR.12		For verifying the certificate (see Certificate Hierarchy) use the rules for X.509 certificates [9]. The Charge Point MUST verify the file's digital signature using the Firmware Signing certificate.	
L01.FR.13	When the Charge Point enters the Download Scheduled state.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status DownloadScheduled.	For example when it is busy with installing another firmware or it is busy Charging.
L01.FR.14	When the Charge Point enters the Download Paused state.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status DownloadPaused.	For example when the Charge Point has tasks with higher priorities.
L01.FR.15	When a Charge Point needs to reboot before installing the downloaded firmware.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status InstallRebooting, before rebooting.	
L01.FR.16	L01.FR.04 AND When installDateTime is set to a future date-time	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status InstallScheduled and install the firmware at the specified installation time.	

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.17	L01.FR.16 AND current DateTime >= InstallDateTime	The Charge Point SHALL install the new firmware as soon as it is able to.	
L01.FR.18	L01.FR.17 AND The Charge Point has ongoing transactions AND It is not possible to continue charging during installation of firmware	The Charge Point SHALL wait until all transactions have ended, before commencing installation.	
L01.FR.19	L01.FR.18	The Charge Point SHALL set all connectors that are not in use to UNAVAILABLE while the Charge Point waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.	
L01.FR.20	When the Charge Point receives a UpdateFirmware.req (the original OCPP 1.6 message)	The Charge Point SHALL respond with a WebSocket RPC CALLERROR NotSupported, and the Charge Point SHALL NOT start the Firmware Update process.	
L01.FR.21		The field requestId in SignedFirmwareStatusNotification.req is mandatory, unless *status* = `Idle`.	
L01.FR.22	When the Charge Point needs to reboot during a firmware update AND the bootloader is unable to send OCPP messages	The Charge Point MAY omit the SignedFirmwareStatusNotification.req (status=Installing) message.	
L01.FR.23	When the Charge Point receives an SignedUpdateFirmware.req	The Charge Point SHALL validate the certificate before accepting the message.	
L01.FR.24	L01.FR.23 AND the certificate is invalid	The Charge Point SHALL respond with SignedUpdateFirmware.conf (status=InvalidCertificate).	
L01.FR.25	L01.FR.23 AND the certificate is revoked	The Charge Point SHALL respond with SignedUpdateFirmware.conf (status=RevokedCertificate).	
L01.FR.26	When a Charge Point is installing new Firmware OR is going to install new Firmware, but has received an SignedUpdateFirmware.req command to install it at a later time AND the Charge Point receives a new SignedUpdateFirmware.req	The Charge Point SHOULD cancel the ongoing firmware update AND respond with status *AcceptedCanceled*.	The Charge Point SHOULD NOT first check if the new firmware file exists, this way the Central System will be able to cancel an ongoing firmware update without starting a new one.

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.27	L01.FR.26 AND the Charge Point is unabled to cancel the installation	The Charge Point MAY respond with status *Rejected*.	
L01.FR.28	Charge Point receives a ExtendedTriggerMessage.req for `FirmwareStatusNotification` AND last sent SignedFirmwareStatusNotification.req had *status* = `Installed`	Charge Point SHALL return a SignedFirmwareStatusNotification.req with *status* = `Idle`.	
L01.FR.29	Charge Point receives a ExtendedTriggerMessage.req for `FirmwareStatusNotification` AND last sent SignedFirmwareStatusNotification.req had *status* <> `Installed`	Charge Point SHALL return a SignedFirmwareStatusNotification.req with the last sent *status*.	

5. Messages

To add the functionality needed for this WhitePaper, a couple of messages have been added from OCPP 2.0.1. Most have their original name from OCPP 2.0.1. Others have a modified name, because they have been modified between 1.6 and 2.0.1. The messages that have been renamed, are marked as such.

5.1. CertificateSigned.req

This contains the field definition of the CertificateSigned.req PDU sent by the Central System to the Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateChain	string[0..10000]	1..1	Required. The signed PEM encoded X.509 certificates. This can also contain the necessary sub CA certificates. The maximum size of this field is be limited by the configuration key: `CertificateSignedMaxSize`.

5.2. CertificateSigned.conf

This contains the field definition of the CertificateSigned.conf PDU sent by the Charge Point to the Central System in response to a CertificateSigned.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	CertificateSignedStatusEnumType	1..1	Required. Returns whether certificate signing has been accepted, otherwise rejected.

5.3. DeleteCertificate.req

Used by the Central System to request deletion of an installed certificate on a Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateHashData	CertificateHashDataType	1..1	Required. Indicates the certificate of which deletion is requested.

5.4. DeleteCertificate.conf

Response to a DeleteCertificate.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	DeleteCertificateStatusEnumType	1..1	Required. Charge Point indicates if it can process the request.

5.5. ExtendedTriggerMessage.req

This contains the field definition of the ExtendedTriggerMessage.req PDU sent by the Central System to the Charge Point.

This message is based on the OCPP 2.0.1 TriggerMessageRequest, it has been renamed to: ExtendedTriggerMessage.req, because the original name conflicts with the TriggerMessage.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
requestedMessage	MessageTriggerEnumType	1..1	Required. Type of the message to be triggered.
connectorId	integer connectorId > 0	0..1	Optional. Only filled in when request applies to a specific connector.

5.6. ExtendedTriggerMessage.conf

This contains the field definition of the ExtendedTriggerMessage.conf PDU sent by the Charge Point to the Central System in response to ExtendedTriggerMessage.req.

This message is based on the OCPP 2.0.1 TriggerMessageResponse, it has been renamed to: ExtendedTriggerMessage.conf, because the original name conflicts with the TriggerMessage.conf from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	TriggerMessageStatusEnumType	1..1	Required. Indicates whether the Charge Point will send the requested notification or not.

5.7. GetInstalledCertificateIds.req

Used by the Central System to request an overview of the installed certificates on a Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateType	CertificateUseEnumType	1..1	Required. Indicates the type of certificates requested.

5.8. GetInstalledCertificateIds.conf

Response to a GetInstalledCertificateIDs.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	GetInstalledCertificateStatusEnumType	1..1	Required. Charge Point indicates if it can process the request.
certificateHashData	CertificateHashDataType	0..*	Optional. The Charge Point includes the Certificate information for each available certificate.

5.9. GetLog.req

This contains the field definition of the GetLog.req PDU sent by the Central System to the Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
logType	LogEnumType	1..1	Required. This contains the type of log file that the Charge Point should send.
requestId	integer	1..1	Required. The Id of this request
retries	integer	0..1	Optional. This specifies how many times the Charge Point must try to upload the log before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charge Point to decide how long to wait between attempts.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
log	LogParametersType	1..1	Required. This field specifies the requested log and the location to which the log should be sent.

5.10. GetLog.conf

This contains the field definition of the GetLog.conf PDU sent by the Charge Point to the Central System in response to a GetLog.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	LogStatusEnumType	1..1	Required. This field indicates whether the Charge Point was able to accept the request.
filename	string[0..255]	0..1	Optional. This contains the name of the log file that will be uploaded. This field is not present when no logging information is available.

5.11. InstallCertificate.req

Used by the Central System to request installation of a certificate on a Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateType	CertificateUseEnumType	1..1	Required. Indicates the certificate type that is sent.
certificate	string[0..5500]	1..1	Required. An PEM encoded X.509 certificate.

5.12. InstallCertificate.conf

The response to a InstallCertificate.req, sent by the Charge Point to the Central System.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	CertificateStatusEnumType	1..1	Required. Charge Point indicates if installation was successful.

5.13. LogStatusNotification.req

This contains the field definition of the LogStatusNotification.req PDU sent by the Charge Point to the Central System.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	UploadLogStatusEnumType	1..1	Required. This contains the status of the log upload.
requestId	integer	0..1	Optional. The request id that was provided in the GetLog.req that started this log upload.

5.14. LogStatusNotification.conf

This contains the field definition of the LogStatusNotification.conf PDU sent by the Central System to the Charge Point in response to LogStatusNotification.req.

No fields are defined.

5.15. SecurityEventNotification.req

Sent by the Charge Point to the Central System in case of a security event.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
type	string[50]	1..1	Required. Type of the security event (See list of currently known security events)
timestamp	dateTime	1..1	Required. Date and time at which the event occurred.
techInfo	string[0..255]	0..1	Additional information about the occurred security event.

5.16. SecurityEventNotification.conf

Sent by the Central System to the Charge Point to confirm the receipt of a SecurityEventNotification.req message.

No fields are defined.

5.17. SignCertificate.req

Sent by the Charge Point to the Central System to request that the Certificate Authority signs the public key into a certificate.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
csr	string[0..5500]	1..1	Required. The Charge Point SHALL send the public key in form of a Certificate Signing Request (CSR) as described in RFC 2986 [14] and then PEM encoded, using the SignCertificate.req message.

5.18. SignCertificate.conf

Sent by the Central System to the Charge Point in response to the SignCertificate.req message.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	GenericStatusEnumType	1..1	Required. Specifies whether the Central System can process the request.

5.19. SignedFirmwareStatusNotification.req

This contains the field definition of the SignedFirmwareStatusNotification.req PDU sent by the Charge Point to the Central System.

This is the OCPP 2.0.1 FirmwareStatusNotificationRequest, it has been renamed to SignedFirmwareStatusNotification.req, because the original name conflicts with the FirmwareStatusNotification.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	FirmwareStatusEnumType	1..1	Required. This contains the progress status of the firmware installation.
requestId	integer	0..1	Optional. The request id that was provided in the SignedUpdateFirmware.req that started this firmware update. This field is mandatory, unless the message was triggered by a TriggerMessage.req or the ExtendedTriggerMessage.req AND there is no firmware update ongoing.

5.20. SignedFirmwareStatusNotification.conf

This contains the field definition of the SignedFirmwareStatusNotification.conf PDU sent by the Central System to the Charge Point in response to a SignedFirmwareStatusNotification.req.

This is the OCPP 2.0.1 FirmwareStatusNotificationResponse, it is renamed to: SignedFirmwareStatusNotification.conf, because the original name conflicts with the FirmwareStatusNotification.conf from OCPP 1.6.

No fields are defined.

5.21. SignedUpdateFirmware.req

This contains the field definition of the SignedUpdateFirmware.req PDU sent by the Central System to the Charge Point.

This is the OCPP 2.0.1 UpdateFirmwareRequest, it is renamed to SignedUpdateFirmware.req, it is renamed because the original name conflicts with the UpdateFirmware.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
retries	integer	0..1	Optional. This specifies how many times Charge Point must try to download the firmware before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charge Point to decide how long to wait between attempts.
requestId	integer	1..1	Required. The Id of this request
firmware	FirmwareType	1..1	Required. Specifies the firmware to be updated on the Charge Point.

5.22. SignedUpdateFirmware.conf

This contains the field definition of the SignedUpdateFirmware.conf PDU sent by the Charge Point to the Central System in response to an SignedUpdateFirmware.req.

This is the OCPP 2.0.1 UpdateFirmwareResponse, it is renamed to SignedUpdateFirmware.conf, it is renamed because the original name conflicts with the UpdateFirmware.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	UpdateFirmwareStatusEnumType	1..1	Required. This field indicates whether the Charge Point was able to accept the request.

6. Datatypes

6.1. CertificateHashDataType

Class

CertificateHashDataType is used by: DeleteCertificate.req, GetInstalledCertificateIds.conf

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
hashAlgorithm	HashAlgorithmEnumType	1..1	Required. Used algorithms for the hashes provided.
issuerNameHash	identifierString[0..128]	1..1	Required. hashed value of the IssuerName.
issuerKeyHash	identifierString[0..128]	1..1	Required. Hashed value of the issuers public key

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
serialNumber	string[0..40]	1..1	Required. The serial number of the certificate.

6.2. CertificateSignedStatusEnumType

Enumeration

CertificateSignedStatusEnumType is used by: CertificateSigned.conf

VALUE	DESCRIPTION
Accepted	Signed certificate is valid.
Rejected	Signed certificate is invalid.

6.3. CertificateStatusEnumType

Enumeration

Status of the certificate.

CertificateStatusEnumType is used by: InstallCertificate.conf

VALUE	DESCRIPTION
Accepted	The installation of the certificate succeeded.
Failed	The certificate is valid and correct, but there is another reason the installation did not succeed.
Rejected	The certificate is invalid and/or incorrect OR the CPO tries to install more certificates than allowed.

6.4. CertificateUseEnumType

Enumeration

CertificateUseEnumType is used by: GetInstalledCertificateIds.req, InstallCertificate.req

VALUE	DESCRIPTION
CentralSystemRootCertificate	Root certificate, used by the CA to sign the Central System and Charge Point certificate.
ManufacturerRootCertificate	Root certificate for verification of the Manufacturer certificate.

6.5. DeleteCertificateStatusEnumType

Enumeration

DeleteCertificateStatusEnumType is used by: DeleteCertificate.conf

VALUE	DESCRIPTION
Accepted	Normal successful completion (no errors).
Failed	Processing failure.
NotFound	Requested resource not found.

6.6. FirmwareStatusEnumType

Enumeration

Status of a firmware download.

A value with "Intermediate state" in the description, is an intermediate state, update process is not finished.

A value with "Failure end state" in the description, is an end state, update process has stopped, update failed.

A value with "Successful end state" in the description, is an end state, update process has stopped, update successful.

FirmwareStatusEnumType is used by: SignedFirmwareStatusNotification.req

VALUE	DESCRIPTION
Downloaded	Intermediate state. New firmware has been downloaded by Charge Point.
DownloadFailed	Failure end state. Charge Point failed to download firmware.
Downloading	Intermediate state. Firmware is being downloaded.
DownloadScheduled	Intermediate state. Downloading of new firmware has been scheduled.
DownloadPaused	Intermediate state. Downloading has been paused.
Idle	Charge Point is not performing firmware update related tasks. Status Idle SHALL only be used as in a SignedFirmwareStatusNotification.req that was triggered by ExtendedTriggerMessage.req.
InstallationFailed	Failure end state. Installation of new firmware has failed.

VALUE	DESCRIPTION
Installing	Intermediate state. Firmware is being installed.
Installed	Successful end state. New firmware has successfully been installed in Charge Point.
InstallRebooting	Intermediate state. Charge Point is about to reboot to activate new firmware. This status MAY be omitted if a reboot is an integral part of the installation and cannot be reported separately.
InstallScheduled	Intermediate state. Installation of the downloaded firmware is scheduled to take place on installDateTime given in SignedUpdateFirmware.req.
InstallVerificationFailed	Failure end state. Verification of the new firmware (e.g. using a checksum or some other means) has failed and installation will not proceed. (Final failure state)
InvalidSignature	Failure end state. The firmware signature is not valid.
SignatureVerified	Intermediate state. Provide signature successfully verified.

6.7. FirmwareType

Class

Represents a copy of the firmware that can be loaded/updated on the Charge Point.

FirmwareType is used by: SignedUpdateFirmware.req

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
location	string[0..512]	1..1	Required. URI defining the origin of the firmware.
retrieveDateTime	dateTime	1..1	Required. Date and time at which the firmware shall be retrieved.
installDateTime	dateTime	0..1	Optional. Date and time at which the firmware shall be installed.
signingCertificate	string[0..5500]	1..1	Required. Certificate with which the firmware was signed. PEM encoded X.509 certificate.
signature	string[0..800]	1..1	Required. Base64 encoded firmware signature.

6.8. GenericStatusEnumType

Enumeration

Generic message response status

VALUE	DESCRIPTION
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.

6.9. GetInstalledCertificateStatusEnumType

Enumeration

GetInstalledCertificateStatusEnumType is used by: GetInstalledCertificateIds.conf

VALUE	DESCRIPTION
Accepted	Normal successful completion (no errors).
NotFound	Requested certificate not found.

6.10. HashAlgorithmEnumType

Enumeration

HashAlgorithmEnumType is used by: CertificateHashDataType

VALUE	DESCRIPTION
SHA256	SHA-256 hash algorithm.
SHA384	SHA-384 hash algorithm.
SHA512	SHA-512 hash algorithm.

6.11. LogEnumType

Enumeration

LogEnumType is used by: GetLog.req

VALUE	DESCRIPTION
DiagnosticsLog	This contains the field definition of a diagnostics log file
SecurityLog	Sent by the Central System to the Charge Point to request that the Charge Point uploads the security log.

6.12. LogParametersType

Class

Class for detailed information the retrieval of logging entries.

LogParametersType is used by: GetLog.req

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
remoteLocation	string[0..512]	1..1	Required. The URL of the location at the remote system where the log should be stored.
oldestTimestamp	dateTime	0..1	Optional. This contains the date and time of the oldest logging information to include in the diagnostics.
latestTimestamp	dateTime	0..1	Optional. This contains the date and time of the latest logging information to include in the diagnostics.

6.13. LogStatusEnumType

Enumeration

LogStatusEnumType is used by: GetLog.conf

VALUE	DESCRIPTION
Accepted	Accepted this log upload. This does not mean the log file is uploaded is successfully, the Charge Point will now start the log file upload.
Rejected	Log update request rejected.
AcceptedCanceled	Accepted this log upload, but in doing this has canceled an ongoing log file upload.

6.14. MessageTriggerEnumType

Enumeration

Type of request to be triggered by trigger messages.

MessageTriggerEnumType is used by: ExtendedTriggerMessage.req

VALUE	DESCRIPTION
BootNotification	To trigger BootNotification.req.

VALUE	DESCRIPTION
LogStatusNotification	To trigger LogStatusNotification.req.
FirmwareStatusNotification	To trigger SignedFirmwareStatusNotification.req (So the status of the secure firmware update introduced in this document).
Heartbeat	To trigger Heartbeat.req.
MeterValues	To trigger MeterValues.req.
SignChargePointCertificate	To trigger a SignCertificate.req with certificateType: ChargePointCertificate.
StatusNotification	To trigger SatusNotification.req.

6.15. TriggerMessageStatusEnumType

Enumeration

TriggerMessageStatusEnumType is used by: ExtendedTriggerMessage.conf

VALUE	DESCRIPTION
Accepted	Requested message will be sent.
Rejected	Requested message will not be sent.
NotImplemented	Requested message cannot be sent because it is either not implemented or unknown.

6.16. UpdateFirmwareStatusEnumType

Enumeration

UpdateFirmwareStatusEnumType is used by: SignedUpdateFirmware.conf

VALUE	DESCRIPTION
Accepted	Accepted this firmware update request. This does not mean the firmware update is successful, the Charge Point will now start the firmware update process.
Rejected	Firmware update request rejected.

VALUE	DESCRIPTION
AcceptedCanceled	Accepted this firmware update request, but in doing this has canceled an ongoing firmware update.
InvalidCertificate	The certificate is invalid.
RevokedCertificate	Failure end state. The Firmware Signing certificate has been revoked.

6.17. UploadLogStatusEnumType

Enumeration

UploadLogStatusEnumType is used by: LogStatusNotification.req

VALUE	DESCRIPTION
BadMessage	A badly formatted packet or other protocol incompatibility was detected.
Idle	The Charge Point is not uploading a log file. Idle SHALL only be used when the message was triggered by a ExtendedTriggerMessage.req.
NotSupportedOperation	The server does not support the operation
PermissionDenied	Insufficient permissions to perform the operation.
Uploaded	File has been uploaded successfully.
UploadFailure	Failed to upload the requested file.
Uploading	File is being uploaded.

7. Configuration Keys

7.1. `AdditionalRootCertificateCheck`

Required/optional	optional
Accessibility	R
Type	boolean

Description	When set to true, only one certificate (plus a temporarily fallback certificate) of certificateType CentralSystemRootCertificate is allowed to be installed at a time. When installing a new Central System Root certificate, the new certificate SHALL replace the old one AND the new Central System Root Certificate MUST be signed by the old Central System Root Certificate it is replacing. This configuration key is required unless only "security profile 1 - Unsecured Transport with Basic Authentication" is implemented. Please note that security profile 1 SHOULD only be used in trusted networks. *Note: When using this additional security mechanism please be aware that the Charge Point needs to perform a full certificate chain verification when the new Central System Root certificate is being installed. However, once the old Central System Root certificate is set as the fallback certificate, the Charge Point needs to perform a partial certificate chain verification when verifying the server certificate during the TLS handshake. Otherwise the verification will fail once the old Central System Root (fallback) certificate is either expired or removed.*

7.2. AuthorizationKey

Required/optional	optional
Accessibility	W
Type	String
Description	The basic authentication password is used for HTTP Basic Authentication, minimal length: 16 bytes. It is strongly advised to be randomly generated binary to get maximal entropy. Hexadecimal represented (20 bytes maximum, represented as a string of up to 40 hexadecimal digits). This configuration key is write-only, so that it cannot be accidentally stored in plaintext by the Central System when it reads out all configuration keys. This configuration key is required unless only "security profile 3 - TLS with client side certificates" is implemented.

7.3. CertificateSignedMaxChainSize

Required/optional	optional
Accessibility	R
Type	integer
Description	This configuration key can be used to limit the size of the 'certificateChain' field from the CertificateSigned.req PDU. The value of this configuration key has a maximum limit of 10.000 characters.

7.4. CertificateStoreMaxLength

Required/optional	optional
Accessibility	R
Type	integer

Description	Maximum number of Root/CA certificates that can be installed in the Charge Point.

7.5. CpoName

Required/optional	optional
Accessibility	RW
Type	String
Description	This configuration key contains CPO name (or an organization trusted by the CPO) as used in the Charge Point Certificate. This is the CPO name that is to be used in a CSR send via: SignCertificate.req

7.6. SecurityProfile

Required/optional	optional
Accessibility	RW
Type	integer
Description	This configuration key is used to set the security profile used by the Charge Point. The value of this configuration key can only be increased to a higher level, not decreased to a lower level, if the Charge Point receives a lower value then currently configured,the Charge Point SHALL Rejected the ChangeConfiguration.req Before accepting the new value, the Charge Point SHALL check if all the prerequisites for the new Security Profile are met, if not, the Charge Point SHALL Rejected the ChangeConfiguration.req. After the security profile was successfully changed, the Charge Point disconnects from the Central System and SHALL reconnect using the new configured Security Profile. Default, when no security profile is yet configured: 0.

8. Security Events

The table below provides a list of security events. Security events that are critical should be pushed to the Central System.

SECURITY EVENT	DESCRIPTION	CRITICAL
FirmwareUpdated	The Charge Point firmware is updated	Yes
FailedToAuthenticateAtCentralSystem	The authentication credentials provided by the Charge Point were rejected by the Central System	No

SECURITY EVENT	DESCRIPTION	CRITICAL
CentralSystemFailedToAuthenticate	The authentication credentials provided by the Central System were rejected by the Charge Point	No
SettingSystemTime	The system time on the Charge Point was changed	Yes
StartupOfTheDevice	The Charge Point has booted	Yes
ResetOrReboot	The Charge Point was rebooted or reset	Yes
SecurityLogWasCleared	The security log was cleared	Yes
ReconfigurationOfSecurityParameters	Security parameters, such as keys or the security profile used, were changed	No
MemoryExhaustion	The Flash or RAM memory of the Charge Point is getting full	Yes
InvalidMessages	The Charge Point has received messages that are not valid OCPP messages, if signed messages, signage invalid/incorrect	No
AttemptedReplayAttacks	The Charge Point has received a replayed message (other than the Central System trying to resend a message because it there was for example a network problem)	No
TamperDetectionActivated	The physical tamper detection sensor was triggered	Yes
InvalidFirmwareSignature	The firmware signature is not valid	No
InvalidFirmwareSigningCertificate	The certificate used to verify the firmware signature is not valid	No
InvalidCentralSystemCertificate	The certificate that the Central System uses was not valid or could not be verified	No
InvalidChargePointCertificate	The certificate sent to the Charge Point using the SignCertificate.conf message is not a valid certificate	No
InvalidTLSVersion	The TLS version used by the Central System is lower than 1.2 and is not allowed by the security specification	No
InvalidTLSCipherSuite	The Central System did only allow connections using TLS cipher suites that are not allowed by the security specification	No

9. Changelog

SECTION / USE CASE	CHANGE
2.3. Unsecured Transport with Basic Authentication Profile	Basic auth example added to remarks.
2.4.1. TLS with Basic Authentication Profile	A00.FR.308 changed. "URL or IP address" changed to "FQDN".
2.4.1. TLS with Basic Authentication Profile	A00.FR.317 changed. Added a note.
2.5.1. TLS with Client Side Certificates Profile	A00.FR.405 changed. "unique identifier" changed to "unique serial number".
2.5.1. TLS with Client Side Certificates Profile	A00.FR.412 changed. "URL" changed to "FQDN".
2.5.1. TLS with Client Side Certificates Profile	A00.FR.429 added.
2.6.1. Certificate Properties	A00.FR.507 changed. Encoding changed from DER, followed by Base64 encoding to PEM.
2.6.1. Certificate Properties	A00.FR.510 changed. "full URL of the endpoint" changed to "FQDN".
2.6.2. Certificate Hierarchy	A00.FR.604, A00.FR.605 removed.
A02/A03	Prerequisite added. "The configuration variable `CpoName` MUST be set."
A02/A03	Error handling added.
A02/A03	A02.FR.03/A03.FR.03 changed. PEM encoding included.
A02/A03	A02.FR.04/A03.FR.04 changed. The dedicated authority server MAY be operated by the CPO.
A05	Error handling and requirements; A05.FR.08, A05.FR.09 added.
L01	Added requirements; L01.FR.21, L01.FR.22, L01.FR.23, L01.FR.24, L01.FR.25, L01.FR.26, L01.FR.27, L01.FR.28, L01.FR.29.
M04	M04.FR.07 added.
M05	M05.FR.05, M05.FR.06, M05.FR.07, M05.FR.08, M05.FR.09 added in v1.1. M05.FR.05, M05.FR.07 removed in v1.2 M05.FR.08, M05.FR.09 changed in v1.2 M05.FR.10, M05.FR.11, M05.FR.12, M05.FR.13 added in v1.2
N01	N01.FR.11, N01.FR.12 added.

SECTION / USE CASE	CHANGE
5.1. CertificateSigned.req	Changes in 'cert' field. Field name changed from 'cert' to 'certificateChain'. Field type changed from string[0..5500] to string[0..10000]. Cardinality changed from 1..* to 1..1. Encoding changed from DER, then Hex encoded into a case insensitive string to PEM.
5.7. GetInstalledCertificateIds.req	'typeOfCertificate' field renamed to 'certificateType'.
5.11. InstallCertificate.req	'certificate' field encoding changed from DER, then Hex encoded into a case insensitive string to PEM.
5.17. SignCertificate.req	'csr' field encoding changed from DER to PEM.
5.13. LogStatusNotification.req	'requestId' field cardinality changed from 1..1 to 0..1
5.15. SecurityEventNotification.req	'techInfo' field added.
5.19. SignedFirmwareStatusNotification.req	'requestId' field cardinality changed from 1..1 to 0..1
6.1. CertificateHashDataType	'issuerKeyHash' field type changed from string[0..128] to identifierString[0..128].
6.1. CertificateHashDataType	'serialNumber' field type changed from string[0..20] to string[0..40].
6.6. FirmwareStatusEnumType	Enum values 'InvalidCertificate', 'RevokedCertificate', 'CertificateVerified' removed.
6.7. FirmwareType	'signingCertificate' field encoding changed from DER, then Hex encoded into a case insensitive string to PEM.
6.16. UpdateFirmwareStatusEnumType	Enum values 'InvalidCertificate', 'RevokedCertificate' added.
7. Configuration Keys	Configuration key 'CertificateSignedMaxChain' removed.
7. Configuration Keys	Configuration key 'CertificateSignedMaxChainSize' added.
7. Configuration Keys	Configuration key 'AdditionalRootCertificateCheck' added.
8. Security Events	'FailedToAuthenticateAtCentral System' changed to: 'FailedToAuthenticateAtCentralSystem' removed incorrect whitespace.
8. Security Events	'Central SystemFailedToAuthenticate' changed to: 'CentralSystemFailedToAuthenticate' removed incorrect whitespace.