OCIT®

Open Communication Interface for Road Traffic Control Systems

Offene Schnittstellen für die Straßenverkehrstechnik

OCIT-Outstations
Profile 4 – VPN
OpenVPN Configuration and related Certificate Management

OCIT-O-Profil_4_V1.0_A01

OCIT   Developer Group (ODG)
OCIT® is a registered trademark of the companies Siemens, SWARCO, AVT-STOYE and Stührenberg

OCIT-Outstations

Profile 4 – VPN

OpenVPN Configuration and related Certificate Management

Document: OCIT-O-Profil_4_V1.0_A01

Issued by: OCIT Developer Group (ODG)

Contact: www.ocit.org

Table of content

History

Version Issue	Distribution to	Date	Comments
V 1.0 A01	PUBLIC	June 11, 2019	Initial Version

Specifications

The OCIT outstations configuration document **OCIT-O CD Vx.x** contains an overview of all of the specifications having a copyright administered by ODG and assigns versions and issue statuses according to:

- associated specifications of the interface "OCIT outstations for traffic signal controllers" with reference to the corresponding OCIT instations and OCIT-C specifications,
- gives information on the use of the transmission profiles and
- provides an overview of packages of specifications for interfaces for the use of which a nominal fee is required by ODG

This document has been released on June 11th 2019. Please note that due to advances in cryptography it is strongly recommended to review and update the document at least after three years. This will ensure a consistent security level for the mandatory to support cryptographic algorithms and cipher suites. References and conformity statements to OCIT Profile 4 shall be made as OCIT-O Profile 4: 2019.

The current issue of the document is published on www.ocit.org.

1 Introduction

The ubiquitous proliferation of packet-based services on the mobile network (GPRS, UMTS, LTE,...) confronts urban infrastructure operators with the increasing need to use components in the communications network over which they do not have complete control. To this end, the OCIT-O Profile 4 - VPN defines the option of building secure tunnels within these (potentially public) network segments, which support secure communication within the OCIT system.

OCIT-O Profile 4 is independent of the network access as long as the network access uses IP. Therefore, it can be used for different network access types, such as UMTS, DSL, or other.

The OCIT–O Profile 4 specifies the set of minimal required interfaces and formats to enable the setup of VPNs in an OCIT system. OpenVPN has been selected to protect the communication between a field device and the central control. The authentication in OpenVPN will be mutual, based on X.509 certificates. The management of the certificates is also part of this specification.

The general approach of connecting an OCIT-O field device (FD) with an OCIT central using a VPN tunnel and the relation to the OSI stack is depicted in the following figure.

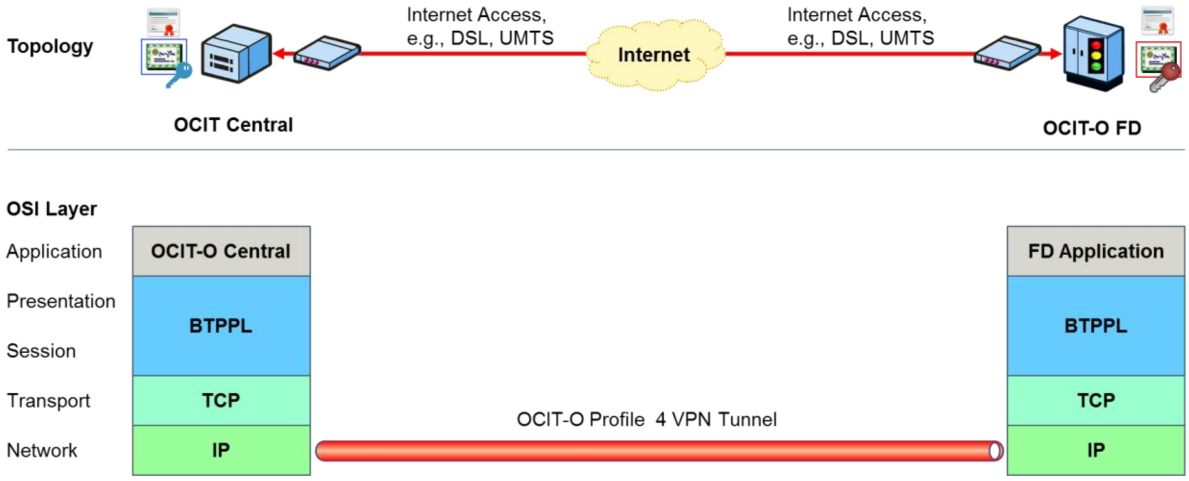

Figure 1 OCIT-O Profil 4 in the ISO-model

1.1 Scope

The specification will describe the minimum set of conventions and interfaces necessary to ensure interoperability between different vendor’s products with respect to VPN application and certificate management. This comprises the interface definition in terms of

- OpenVPN setup and configuration
- Format definitions for the application of certificates (Certificate Signing Requests)
- Format definitions for the certificates and Certificate Revocation Lists (CRL)
- Definition of the minimum content of certificates utilized in OCIT-O Profile 4
- Recommendations for operation

This specification constitutes a part of an overall security architecture.

The security requirements for an overall security architecture, including technical and organizational means, are expected to be defined by the operator of a traffic management solution using a risk-based approach. An integrator will provide a traffic management solution coping with the security requirements based on products supporting the necessary security features. The implemented security means and also deviations are to be documented by the integrator. The documentation enables the operator to verify the compliance of the provided solution with the given security requirements. The operator is responsible for the security of the traffic management solution.

The approval of a traffic management solution does not constitute certification and therefore does not release the installer or operator of the traffic management network from his responsibility for the installed technology and the secure operation of the traffic management solution.

1.2 Out of Scope

This part of the specification will not describe vendor specific implementations options on how to interact with the specified interfaces.

The definition of the access media type is not part of this specification.

OCIT-O Profile 4 is not intended to secure either OCIT-O Profile 1 or OCIT-O Profile 2 connections.

1.3 Requirements Terminology

OCIT-O Profile 4 follows the approach as defined in RFC 2119 [1]. Here, a subset of the defined terminology is used:

- SHALL: This word, or the terms "REQUIRED" or "MUST", mean that the definition is an absolute requirement of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

1.4 Note

Field devices implementing other OCIT-O Profiles cannot be guaranteed to also be capable to support OCIT-O Profile 4. If they can be upgraded with this functionality strongly depends on the utilized operating system and the underlying hardware.

2 VPN Configuration

OCIT profile 4 is implemented using OpenVPN as a basis technology. Certificate based authentication is used to establish tunnels.

Note: The information described in the following subsections targets the interoperability of different vendor's implementations.

OpenVPN configuration files which incorporate the settings as well as further recommended settings and an argumentation for the settings will be provided by the ODG to ODG licensees upon request.

2.1 General Requirements

2.1.1 Security Events

Throughout the document there are security events specified. These security events relate to potential errors or warnings occurred during the verification of security parameters. Implementations SHOULD provide a mechanism for announcing security events.

Note: potential mechanisms may comprise inherent options of OCIT-O or application additional protocols like syslog or SNMP. In any case, it is recommended to log security events.

2.1.2 Connection Establishment

The minimum version of TLS to be supported is TLS 1.2. The proposal of versions prior to TLS 1.2 should raise a security event ("incident: unsecure communication"). Implementations should provide a mechanism for announcing security events.

2.1.3 BTPPL timeouts

The BTPPL timeouts used in OCIT Profile 4 are identical to the ones defined in OCIT-O profile 1 (see also OCIT-O protocol, chapter 5.3.1 Timeout).

2.2 Requirements for the Control Center

In order to meet the requirements of profile 4, the control center must have at least one OpenVPN server with at least version 2.4 The OpenVPN server SHALL be accessible to the field device via an IP connection. In case the field devices are connected to the public internet, the OpenVPN server of the central system SHALL have a public IP address.

2.2.1 Configuration Information transmitted via OpenVPN

The following table shows example configuration parameter for a field device, provided by the central.

The following table shows the minimum parameter to be supported by the central side. Future versions of this specification may require further parameter required by the central side.

If parameters are required by the central side, the client SHALL act accordingly.

Parameter	OpenVPN	Remark / Example

Name	Options	
IP address of the field device		Example: 192.5.16.3
Routes	push "route <NETWORK> <NETMASK>"	Example: 172.5.14.0 255.255.254.0 Routes are either pushed directly or by using the redirect-gateway option.
DNS Server	push "dhcp-option DNS <DNS-IP>"	At least 2 IP addresses for DNS Servers are transmitted (as in profile 1). If only one server is available, both addresses are identical. Example: 172.5.14.254, 172.5.14.254

2.2.2 OCIT Central System Access

Routing and DNS-configuration of the central system has to be designed in such a way, that the field devices connected with OCIT-O profile 4 are reachable via the central system access. Note that the open ports necessary for central system access have been defined in section 2.4.3. These parameters comply with reference to OCIT-O Base Specification, Section 3.1 (see [27]).

2.2.3 Recognition of OpenVPN Tunnel State

The tun-device of the OpenVPN server SHALL be used to determine OpenVPN communication state.

The detection of an error in the tunnel state SHOULD raise a security event ("warning: tunnel state error").

2.3 Requirements for the field device

2.3.1 General

Field devices should not be reachable from outside the VPN tunnel for remote communication.
Field devices complying with this specification SHALL configure the parameter set (see section 2.2.1) received from the central upon receiving.

2.3.2 OpenVPN client

Field devices complying with this specification SHALL have an OpenVPN client supporting certificate based authentication.

Note that the minimum OpenVPN version with ECDSA support is version 2.4 (see also A.1)

2.3.3 OpenVPN client configuration of the field device

The VPN gateway and the corresponding port SHALL be configurable.

2.3.4 Recognition of OpenVPN tunnel state

The tun-device of the OpenVPN client SHOULD be used to determine OpenVPN communication state.

The detection of an error in the tunnel state SHOULD raise a security event ("warning: tunnel state error").

2.4 Requirements for the network infrastructure

2.4.1 Reachability

To enable the use of OpenVPN in an operator network, certain conditions have to be met.

- Firewalls on the communication path SHALL be configured to allow OpenVPN traffic with the appropriate rules. Note that the communication path may also comprise the mobile network and thus require support of mobile network operator.
- The OpenVPN gateway SHOULD be addressable using a FQDN. This requires that the field device is provisioned with a network address and a DNS address for the OpenVPN tunnel establishment. The OpenVPN gateway SHOULD use a fixed IP address.

2.4.2 Quality of service requirements for the network attachment

For the IP connection from the device to the central system (and back) the following conditions apply:

- Leased lines are used, as experience has shown that dial-up connections (dial-on-demand technologies) do not allow stable communication.
- Effective data rate SHALL be at least 28.800 bit/s in each direction (as with OCIT-O via V.34 lines).

2.4.3 Required services for OCIT-O communication

The following table describes the required services between central ⬄ field device through the VPN tunnel. Note that this needs to be considered for the firewall filtering the traffic received via the OpenVPN tunnel.

Service	from	Connection establishment	to	Protocol	Receiver Port(s)	Remarks
btppl	Central	→ ←	Field device	tcp	2504, 3110, 5001	
dns	Central	←	Field device	udp	53	
	Central	←	Field device	tcp	53	
ntp	Central	←	Field device	udp	123	
n.a.	Central	→ ←	Field device	icmp	n.a.	Required for “ping”, diagnostic measures

Note that additional ports may be necessary to be opened depending on the requested additional services. These additional open ports SHOULD be documented.

2.5 Cryptographic parameter configuration for OpenVPN

In OpenVPN there is a distinction between the control channel and the data channel. For the control channel a TLS cipher suite is negotiated (and therefore a carefully selected set of cipher suites need to be configured), while for the data channel a separate list of ciphers needs to be provided.

2.5.1 Selection of TLS cipher suites for the control channel

The following TLS cipher suites have been selected for support in OCIT-O Profile 4 for the security of the control channel. Annex A.1 lists the mandatory support of cipher suites on client and server site.

Key exchange		Encryption	Hash	Source	
Algorithm	Signature				
TLS_ECDHE_	ECDSA_	WITH_AES_128_GCM_	SHA256	RFC 5289	
TLS_ECDHE_	ECDSA_	WITH_AES_128_CBC_	SHA256	RFC 5289	
TLS_ECDHE_	RSA_	WITH_AES_128_GCM_	SHA256	RFC 5289	
TLS_ECDHE_	RSA_	WITH_AES_128_CBC_	SHA256	RFC 5289	
TLS_DHE_	RSA_	WITH_AES_128_GCM_	SHA256	RFC 5288	
TLS_DHE_	RSA_	WITH_AES_128_CBC_	SHA256	RFC 5246	

Note that the order of cipher suites reflects the prioritization for the finally selected cipher suite.

To support also signatures algorithms with SHA 256 or higher, the `signature_algorithm` extension of TLS[25] SHALL be used. Note that TLS 1.2 is used as protocol version. The `signature_algorithm` extension SHALL state the allowed combinations of hash and signature algorithms.

In the following recommendations for the implementation are provided:

- Implementations using OpenSSL [23]
  - The elliptic curve *secp256r1*is named *prime256v1.*
  - Starting from OpenSSL version 1.0.2, Brainpool curves are supported intrinsically.
- Implementations using OpenVPN [24]
  - Support for ECC based signatures is provided starting from OpenVPN 2.4
    - The selection of cipher suites in operation needs to be done based on the capabilities of the implementation and should also take the installed key material into account. Specifically, if OpenVPN versions smaller than version 2.4 are used, elliptic curve based cipher suites may not be offered in the handshake from the client (field device). Appendix A.2 lists the mandatory support of cipher suites on client and server site.
  - Configuration of cipher lists: Starting from OpenVPN 2.4 the support of cipher lists is provided. In contrast to earlier versions it is possible to define a list of supported cipher suites (ncp-cipher) instead of just a single cipher (cipher). The cipher SHALL be aligned with the cipher suites stated in 2.5.2.

2.5.2 Selection of cipher suites for the data channel

A cipher suite for the OpenVPN data channel is determined by two parameters, the cipher and the authentication algorithm. Note that the authentication parameter needs to be explicitly defined, as the fall back value is SHA-1, which is known to have weaknesses. This is specifically important for all ciphers using different modes than Galois Counter Mode (GCM). GCM realizes authenticated encryption and performs authentication and encryption simultaneously.

The following cipher and authentication parameter have been selected for support in OCIT-O Profile 4 for the security of the data channel. Annex A.1 lists the mandatory support of ciphers on client and server site.

Starting from version 2.4 OpenVPN supports two different notions of ciphers in the configuration:

- *cipher*: determines the actual cipher to be used.
- *ncp-cipher*: list of acceptable ciphers instead of a single option. Allows for instance the support of different clients.

cipher / ncp-cipher	Auth	Note
`AES-256-GCM`	-	Authentication will be done as part of AEAD
`AES-256-CBC`	SHA256	
`AES-128-GCM`	-	Authentication will be done as part of AEAD
`AES-128-CBC`	SHA256	

2.6 Certificate validation

The following subsections describe the certificate validation. The certification validation SHALL be performed by the client and the server and SHALL consider the certificate policy as well as the complete certificate path till the locally stored root CA certificate.

2.6.1 Certificate availability

Certificates shall be used to establish an OpenVPN session by both the server and the client.

The connection termination due to the lack of a certificate of either side SHOULD raise a security event ("warning: certificate unavailable").

The failure of a matching CA issued certificate SHOULD raise a security event ("warning: CA not found").

2.6.2 Certificate policy

Certificates policy SHALL be validated by both the server and the client. The certificate validation SHALL include the mandatory defined fields according to the certificate policy regarding match of allowed / expected values. The mandatory certificate fields and extensions for OCIT-O Profile 4 are defined in section 3.4.

An error validating any of the mandatory certificate field or extensions SHALL not be accepted and SHOULD raise a security event ("incident: certificate parameter wrong").

2.6.3 Certificate revocation

Certificate revocation shall follow the mandatory parameters and procedures specified in ISO/IEC 9594-8.

The management of the Certificate Revocation List (CRL) is a local implementation issue.

An implementation claiming conformance to this standard shall be capable of checking the local CRL at a configurable interval.

The inaccessibility of a CRL SHOULD raise a security warning ("warning: CRL not accessible ").

The expiry of a CRL shall raise a security event ("warning: CRL expired").

Revoked certificates shall not be used or accepted in the establishment or renegotiation of an OpenVPN session. An entity receiving a revoked certificate during session establishment shall refuse the connection. An entity receiving a revoked certificate during session renegotiation shall terminate the connection. The central site (server) SHALL always perform the revocation check as part of the certificate validation.

The field device site (client) SHOULD check the revocation state, if available. As reasoning for the optional check on the client site is the higher exposure of the field device. Through this higher exposure, a key compromise is more likely requiring the verification of the server side.

The refusal / termination of a connection due to a revoked certificate SHOULD raise a security event ("incident: revoked certificate").

2.6.4 Certificate expiry

Expired certificates SHALL not be used or accepted in the establishment or renegotiation of an OpenVPN session. An entity receiving an expired certificate during session establishment SHALL refuse the connection. An entity receiving an expired certificate during session renegotiation shall terminate the connection.

The refusal of a connection due to an expired certificate SHOULD raise a security event ("warning: expired certificate").

2.7 Additional Error handling

Additionally to the security events already defined in the previous sections the following list contains potential errors, which may occur during the OpenSSL handshake.

This information may be adopted by the ODG in the “operational messages”.

- The termination of a connection due to a missed session renegotiation should raise a security event ("incident: session renegotiation interval expired"). Implementations should provide a mechanism for announcing security events.
- The proposal of versions prior to TLS 1.2 should raise a security event ("incident: unsecure communication"). Implementations should provide a mechanism for announcing security events.

3 Certificate Management

3.1 Public Key Infrastructure (informative)

This section provides an overview about general concepts of a Public Key Infrastructure (PKI). This information is provided here as a PKI provides all means to manage certificates. As OCIT-O Profile 4 involves the application of certificates in the context of openVPN, the certificate management is addressed in this document. In general, PKI services are defined in RFC 5280 [2].

3.1.1 General Concept of Certificates

According to FIPS 186-4 [3] a certificate is a set of data that uniquely identifies a key pair and an owner that is authorized to use the key pair. The certificate contains the owner's public key and possibly other information and is digitally signed by a Certification Authority or CA (i.e., a trusted party), thereby binding the public key to the ID of the owner. A certificate is public and may be see similar to an ID or a passport.

Note that in the context of this specification the focus is placed on X.509 certificates issued by a CA (or subordinate CA) providing a common root of trust for all participating entities.

The contained public key is part of an asymmetric key pair and has a corresponding private key. Figure 2 depicts the general concept of public/private keys and certificates. The corresponding private key is crucial as it is used to create digital signatures, which in turn can be verified using the certificate. Therefore, access to the private key must be protected accordingly.

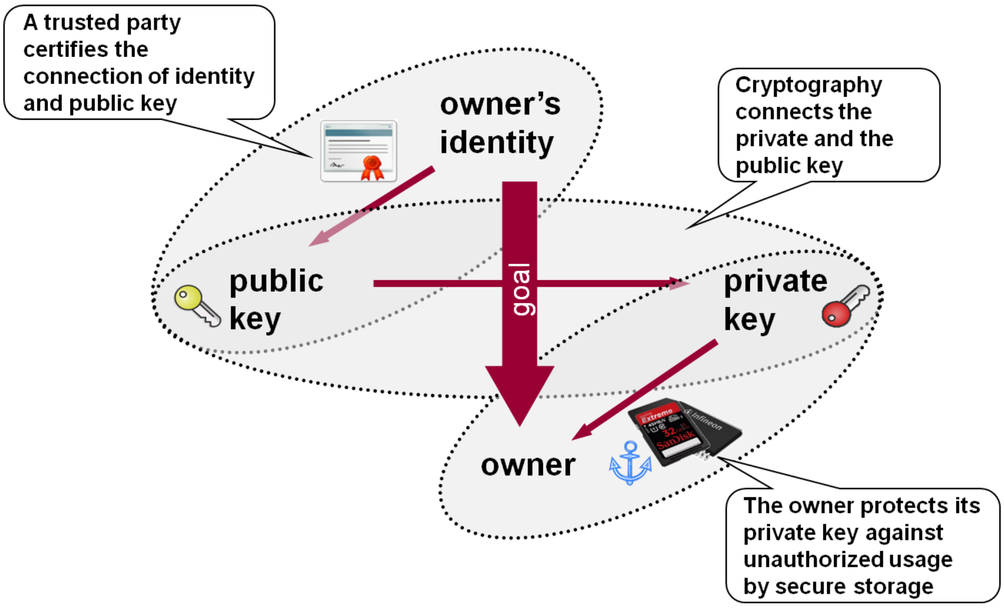
A trusted party certifies the connection of identity and public key
owner's identity
Cryptography connects the private and the public key
public key
goal
private key
owner
The owner protects its private key against unauthorized usage by secure storage

Figure 2 General concept of certificates and public/private key pairs

Certificates and corresponding private keys are used in a variety of (security) protocols for entity authentication and also in the context of session key management. OpenVPN is one of those protocols and is the target for defining the protection of the communication between field device and a central control center in the context of OCIT-O Profile 4.

Note that entity authentication can refer to physical entities like a field device or humans; depending on the context the key pair is used.

3.1.2 Certificate Lifecycle

Certificates and corresponding private keys have a lifecycle, which is depicted in Figure 3 below.

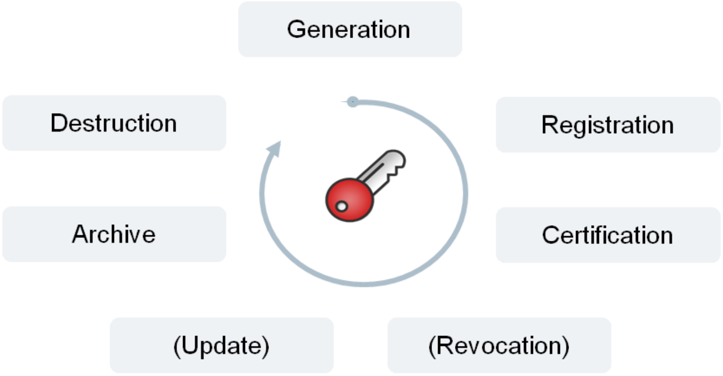

Figure 3: Certificate lifecycle

The following list briefly describes the single stages.

- **Generation**: Asymmetric key pairs, containing a public key and a corresponding private key, can be created by the entity itself or via a central authority (e.g., the CA). To generate cryptographic keys, a good source of entropy is needed to ensure the required randomness of the generated key material.
  Note that for OCIT-O Profile 4 the focus is on local key generation.
- **Registration:** The registration through a Registration Authority (RA) verifies the "identity" of an entity. This is typically done by a Certificate Signing Request (CSR), generated by the applying entity. After identity verification and authorization checks, e.g. with data from an inventory, engineering system or user directory, the RA provides the certificate request to the Certification Authority (CA). This step is also known as enrollment.
- **Certification**: Based on the CSR, the CA generates a digitally signed certificate
- **Revocation**: Certificate revocation may occur when a key pair is no longer authorized to be used, even the certificate has not expired, yet. This may be done for instance if the private key of an entity has been compromised in case of component certificates or if a user leaves the company in case of user related certificates.
- **Update**: Cryptographic keys need to be regularly updated. Cryptographic keys have a dedicated lifetime, e.g., user certificates typically have a lifetime of 2 years, while web server certificates are typically limited to 1 year. Note that there are also long lived certificates, which may have a lifetime of 10 years and above.
- **Archiving**: Public key certificates are recommended to be stored centrally to support signature verification, even after the certificates have expired. This may be necessary for audit purposes.
- **Destruction**: When a component is decommissioned, the cryptographic key material on that component should be securely deleted to avoid any misuse of this key material.

3.1.3 **Supporting Infrastructure**

Certificates are typically managed using a PKI. A PKI has different functionalities and responsibilities. Technically, the components of a PKI are defined within IETF RFC 5280. The interaction of the components is depicted in Figure 4 below.

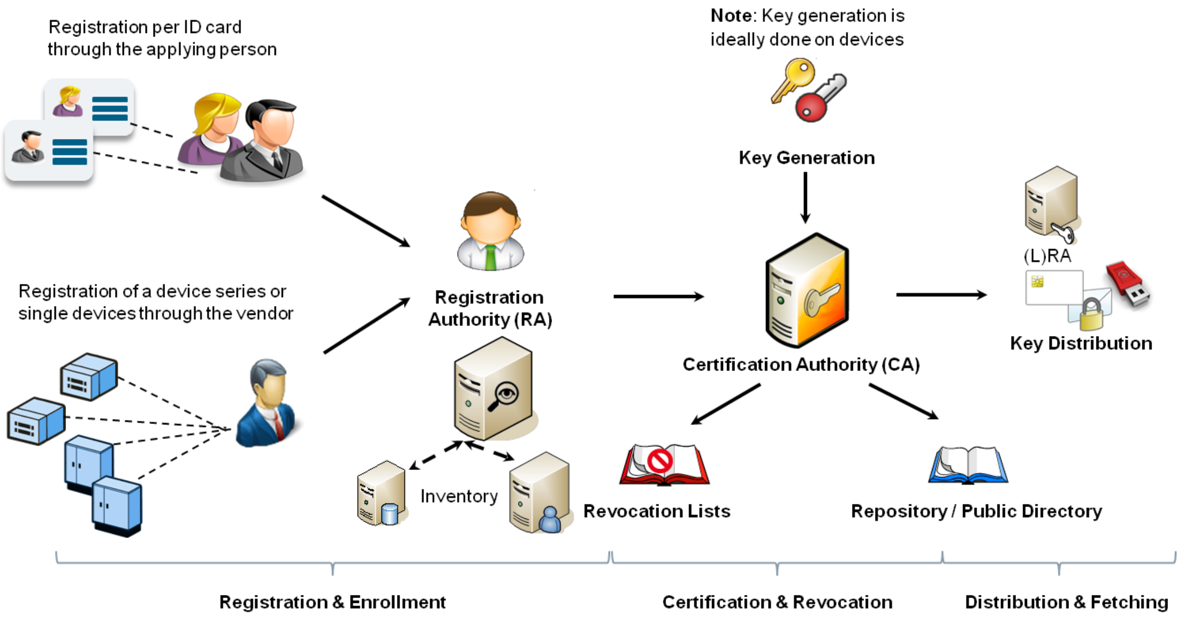
Registration per ID card through the applying person
Note: Key generation is ideally done on devices
Key Generation
Registration Authority (RA)
Certification Authority (CA)
(L)RA
Key Distribution
Registration of a device series or single devices through the vendor
Inventory
Revocation Lists
Repository / Public Directory
Registration & Enrollment
Certification & Revocation
Distribution & Fetching

Figure 4: Overview of Public Key Infrastructure and examples for interactions

Figure 4 also shows the different tasks in the context of the certificate lifecycle (see also section 3.1.2) are performed by the PKI.

Besides the depicted tasks, there are several organizational to determine for the PKI application in a specific context. This is shown in Figure 5 below.

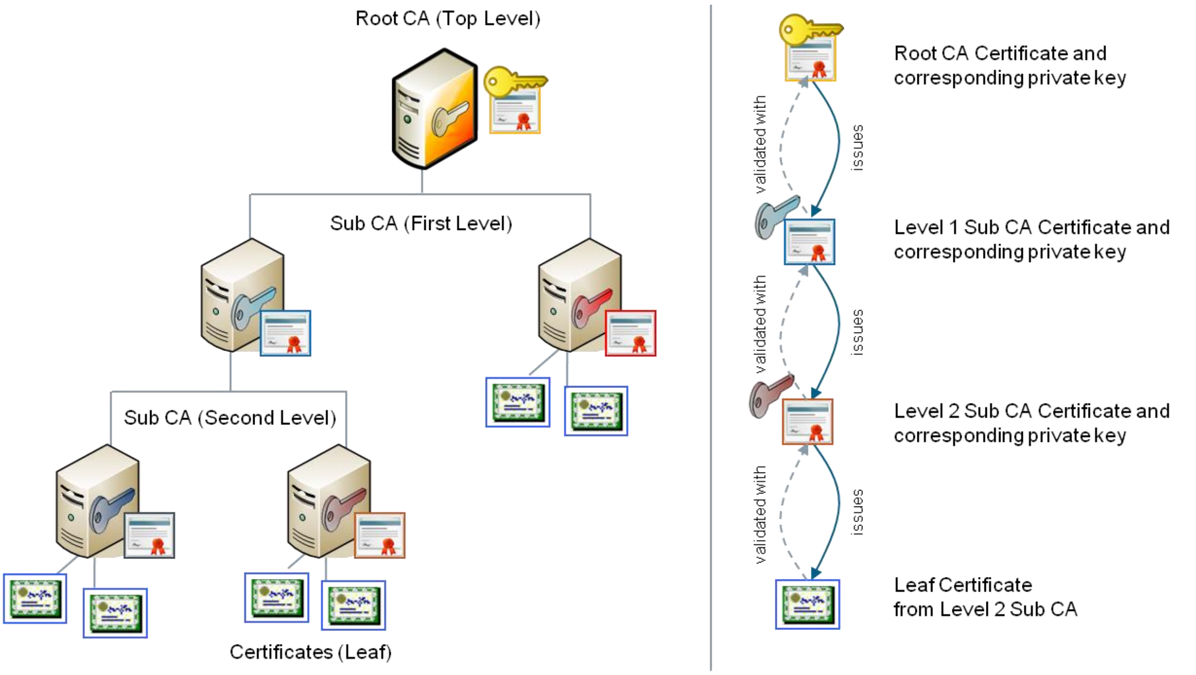
Root CA (Top Level)
Sub CA (First Level)
Sub CA (Second Level)
Certificates (Leaf)
Root CA Certificate and corresponding private key
validated with
issues
Level 1 Sub CA Certificate and corresponding private key
validated with
issues
Level 2 Sub CA Certificate and corresponding private key
validated with
issues
Leaf Certificate from Level 2 Sub CA

Figure 5: PKI hierarchy and certification path

Note that Figure 5 depicts the PKI hierarchy and certification path in an abstract way. In the setup shown the RootCA is not directly used to issue certificates, but to issue certificates for subordinate CAs (SubCAs), which may be necessary to better distinguish between different operational tasks or structures. These SubCAs may have further SubCAs, depending on the target environment. The CA issuing leaf or end entity certificates is often called Issuing CA to differentiate from other SubCAs. To be considered for the operational entities is that during a certificate validation, the complete certification path has to be validated, not only the leaf certificate. In Figure 5 this results in the validation of the leaf certificate and the two SubCA certificates in the certificate chain. The RootCA certificate is typically stored on the end entities as trust anchor.

3.2 OCIT-O Profile 4 target scenario (informative)

Note, this section describes the target scenario, in which certificates and corresponding private keys are to be used. This section may be deleted, if there is a joint document describing the application of OpenVPN to connect outstations with the control center.

The application of OpenVPN targets the protection of the information exchange over arbitrary protocols between the ODG central site and the ODG outstations.

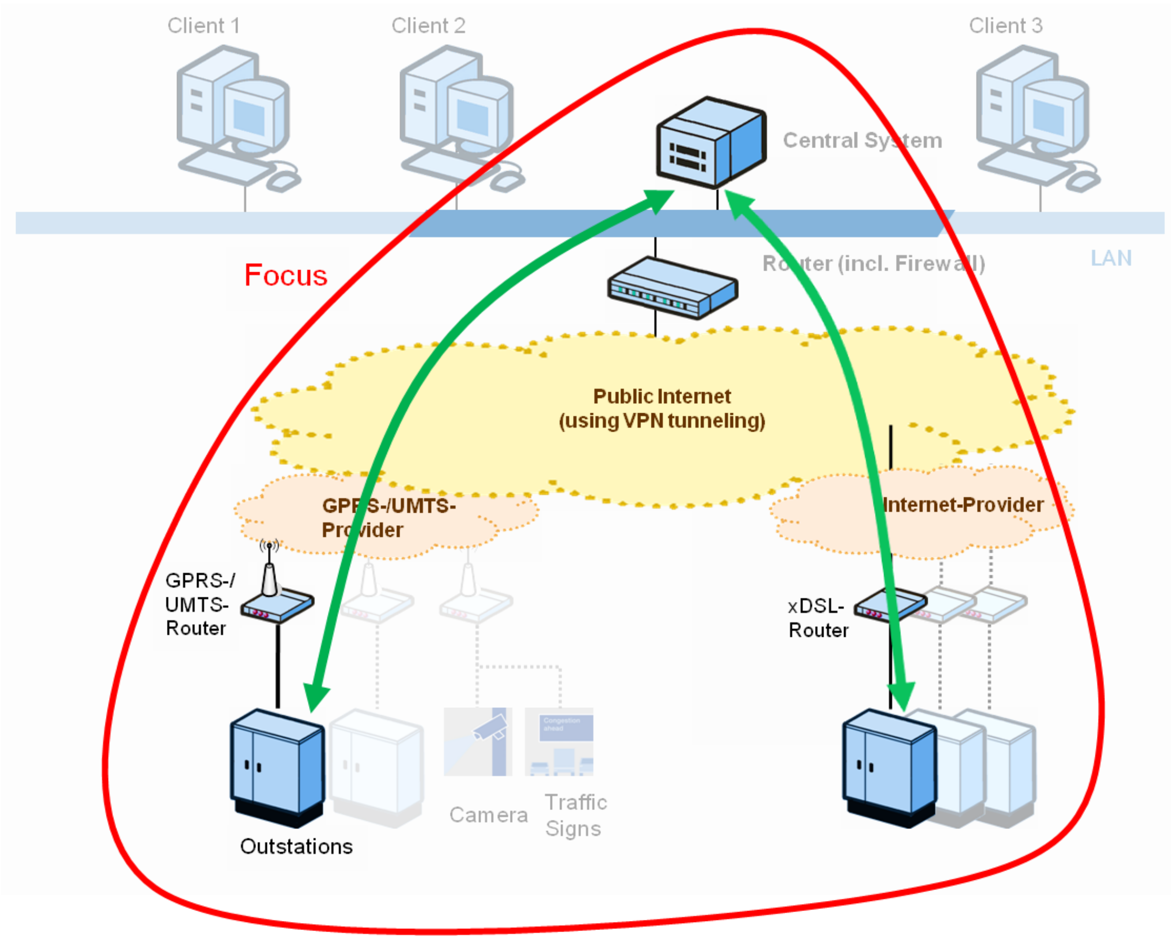

Figure 6: OCIT-O Profile 4 target scenario

The application of OpenVPN in this context enables a network coupling between the outstation and the central site as shown in Figure 6. The termination of the OpenVPN tunnel is intended to be either at the router or directly on the outstation. Note that the termination on the outstation enables an end-to-end device secure channel independent of the utilized connector (e.g., GPRS router, ADSL router, etc.).

To enable such a solution, the following boundary conditions have to be met:

- Availability of X.509 certificate and private key material on all involved components, to enable entity authentication and support of the key negotiation phase of OpenVPN.
- Definition of minimum requirements to the information exchange for enrollment, update, and revocation. This document specifies the exchange format and a certificate profile.
- Procedural support for the connectivity to a PKI managing the X.509 key material (for enrollment, update, and revocation) including a security policy for operation. This is not defined in this document and should be defined from the operator of an ODG solution, utilizing the interfaces defined in this document.

3.3 Certificate Management for OCIT-O Profile 4

The following description aligns with the key pair lifecycle as depicted in Figure 3 and describes the single steps in the context of OCIT-ODG.

3.3.1 Key Pair Generation

Entities performing asymmetric cryptographic functions must possess at least one pair of asymmetric keys. These entities shall be able to generate their own asymmetric key pair. Requirements for random number generation are provided by ISO [13], NIST [14], by the German BSI [15], and ENISA [16] and are recommended to be followed for the integration.

Note that the central generation of key material has been discussed and neglected, as the outstations are expected to be capable of generating their own key material (performance of equipment and available entropy through device internal processes, states and interactions with external environment are considered sufficient).

The following algorithms shall be supported:

- ECDSA with a minimum key length of 256 bit, with supported curves
  - secp256r1 (defined in [20])
  - BrainpoolP256r1 (defined in[21] and for TLS in [22])
- RSA with a minimum key length of 2048 bit

(Note that according to BSI TR 02102-1[17], the length of 2048 bit is accepted as minimum till end of 2022).

To be compliant with this specification, a central entity has to support both algorithms; a field device has to support one of the stated algorithms.

The selection of these algorithms and parameters has been done based on recommendations of the German BSI (cf. [17] and [18]), the NIST (cf. [19]), and the ENISA (cf. [16]). Note also that the support of two mandatory different algorithms and also two different curves for ECDSA allows for immediate reaction if security vulnerabilities are discovered within one crypto system. Moreover, supporting also RSA provides the option to further utilize components already in the field, which may not be upgrades to utilize elliptic curve based cryptography.

The distinct algorithm selection is expected to be done by the operator, based on the specification for a dedicated deployment.

In implementations, secure coding, performance, memory usage, and side-channel attack resistance have an increasingly important role. The algorithms described in this document have been carefully selected to allow patent-free and/or license-free implementations. Nevertheless, some of the described algorithms or its particular implementations may be subject of patent rights. The ODG shall not be held responsible for identifying any or all such patent rights.

3.3.2 Registration and Enrollment

After the key pair generation, the public key of the entity needs to be registered at the PKI. This process depends on the key generation and is described in the following based on an entity local key generation. Once the key is registered, the CA may issue a certificate for that public key depending on the authorization. After certification, the certificate is provided to the entity, which is then enrolled at that specific CA.

The enrollment described in the following provides an example interaction of a service technician and the enrolling end device and the operator infrastructure. Note that there may be deviations in the practical deployment, depending on device characteristics. The description here is intended to provide an overview about necessary steps in the enrollment and serves as a base for the specification of the exchange formats later on.

The process is described in a stepwise approach as shown in Figure 7. Note that the RA/CA component shown in this figure is shown as a combination of a Registration Authority and a Certification Authority. In practice, these may be different entities. Also, in practice the CA is a subordinate CA. To perform the enrollment as depicted there are certain prerequisites. These are listed in the following:

- Trust to a specific RootCA as trust anchor is defined in the operational process. The RootCA certificate as trust anchor is configured on the considered components.
- The control center owns a valid certificate with corresponding private key. The certificate can be traced back to the RootCA.
- The service technician and the control center have a mutual trust.
- The SW tool of the service technician acts as local registration authority (LRA) and the service technician as the LRA officer. The service technician authenticates the field devices as devices to be installed and configured at the specific location (e.g., based on a device identifier).

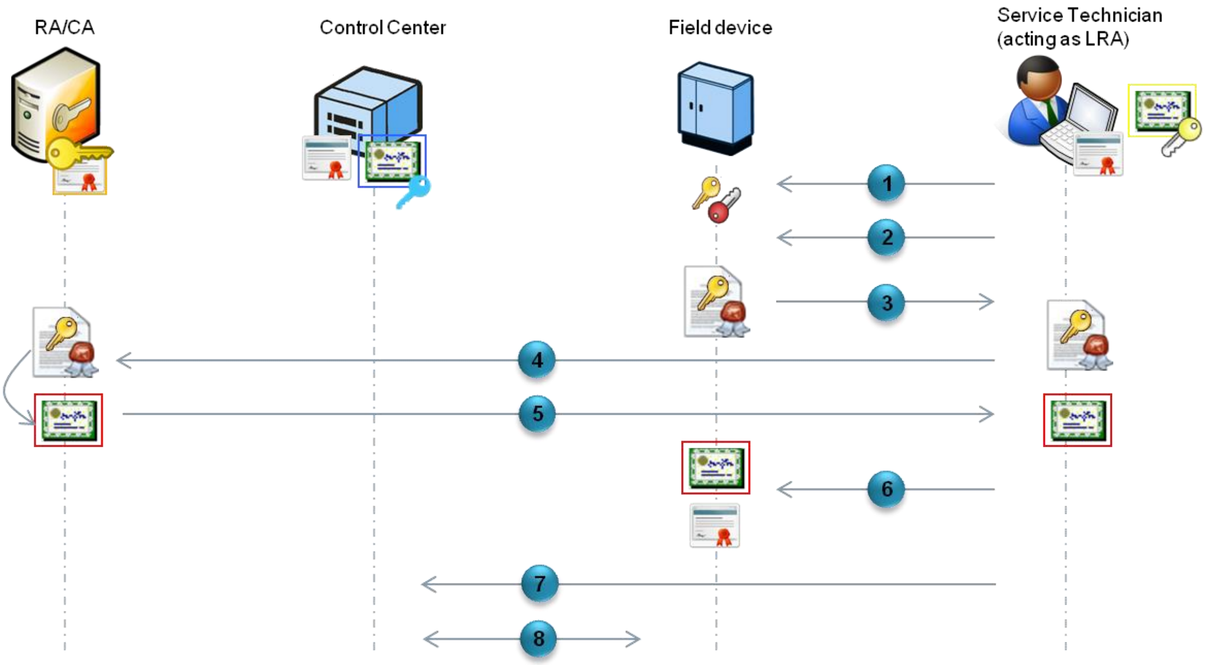

Figure 7: Registration and enrollment of an entity at the PKI

The sequence shown in Figure 7 comprises the following steps. Note that the steps do relate to PKI specific tasks only. How the service technician logs on to the device is out of scope for the description:

1. Service technician performs login to field device and initiates the key pair generation (alternatively key pair may be generated on first power up automatically) according to control center requirements (algorithm, key length, etc.)
2. Upon finishing the key pair generation, the service technician initiates the generation of a Certificate Signing Request (CSR, as specified in RFC 2986 - PKCS #10). For this, additional information may be needed like the ZNr and the FNr and the operator domain to build the `Subject` component or other.

3. Upon finishing the CSR generation, the service technician exports the CSR into a file (extension typically named .csr) and imports the CSR onto his service laptop.
4. Service technician (as authorized LRA officer) sends CSR to a (specific) RA/CA to apply for the certificate
5. RA/CA verifies the service technician's authorization (means for authorization depends on the operators security policy) and potentially against further authorization information (e.g., from the inventory management system) and in the success case creates the certificate and grant access to it (format e.g. DER or Base-64 encoded binary, PKCS#7). In case of an authorization error a log entry should be generated.
6. The service technician imports the certificate file (and maybe certificate path) to the appropriate end entity, includes RootCA and potentially the address of the control center (as in the unsecured case).

The following two steps relate rather to the operation after the enrollment has been finished.

7. The service technician provides the information about relation of the enrolled field device (address) and FNr and ZNr (contained in the certificate) as well as the operator domain to the control center for authentication during OpenVPN connection establishment.
8. After update of system configuration a mutual authenticated VPN tunnel between control center and field device can be established.

3.3.3 **Revocation**

Revocation of certificates may become necessary for instances when the private key of a certificate has been compromised during the normal validity period of the certificate or when the component is taken out of service before the end of the validity period of the certificate. The revocation is typically being performed by the issuing CA either based upon a local operative action or based upon a notification from the device itself (only feasible in conjunction with automated installation). This information is stored in a Certificate Revocation Lists (CRL), for which the format is defined as part of RFC 5280 [2].

A CRL typically contains information about the revoked certificate in terms of the certificate serial number and the revocation time at least. Similar to a certificate, a CRL has a validity period and is signed by the issuing CA. A device may fetch a CRL based on the information in the certificate or based on configuration means. Typically, a CRL is downloaded from a HTTP location by an entity validation a certificate. This location may be specified in the CRL Distribution Point (CDP) extension of the certificate.

The revocation state of a certificate is checked during the validation of a received peer certificate, e.g., during the connection establishment of an OpenVPN connection.

To comply with this specification, revocation verification shall be supported on the server side (central side), at minimum with CRLs. The client (field device) may check the CRL if provided. The requirement for checking the revocation state of certificates on the central side bases on the assumption that the probability of compromised private keys is higher on the field device side, as the protection (also physically) is expected to be lower as on the central side. Therefore, it was concluded that the central side has a higher need to check the revocation status.

3.3.4 **Certificate Update**

The certificate update follows a similar approach as the initial enrollment. If the certificate is contained in an inventory system in the backend, the inventory needs to be update accordingly.

3.4 Format Specifications for OCIT-O Profile 4

This section specifies the necessary exchange formats for the management of certificates to ensure interoperability. This comprises the certificate encoding including a minimum set of expected fields or values in, the information exchange for certificate application as well as the revocation information.

Note: It is strongly recommended to create a certificate policy (see IETF RFC 3647 or ISO/IEC 9594-8 as examples).

3.4.1 **Public Key Certificates –Basic Structure**

This section describes required components of an X.509 certificate and their potential recommended settings. Note that Annex 0 describes an example for a certificate profile for different entities of the system is provided.

The utilized public key certificates shall be conformant to X.509 (cf. IETF RFC 5280 [2], see also ITU-T X.509 [4] or ISO/IEC 9594-8 [5]).

The following ASN.1 structure is taken from RFC 5280 and provides an overview about the context:

```
Certificate  ::=  SEQUENCE  {
     tbsCertificate       TBSCertificate,
     signatureAlgorithm   AlgorithmIdentifier,
     signature            BIT STRING
}

TBSCertificate  ::=  SEQUENCE  {
     version         [0]  Version DEFAULT v1,
     serialNumber         CertificateSerialNumber,
     signature            AlgorithmIdentifier,
     issuer               Name,
     validity             Validity,
     subject              Name,
     subjectPublicKeyInfo SubjectPublicKeyInfo,
     issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                          -- If present, version MUST be v2 or v3
     subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                          -- If present, version MUST be v2 or v3
     extensions      [3]  Extensions OPTIONAL
                          -- If present, version MUST be v3 --
}
```

The following subsections specify specific settings of the `TBSCertificate` components to be used in OCIT-O Profile 4. For all other, the description within RFC 5280 [2] is referred to.

Upon certificate usage, the fields of the certificate are to be validated by the receiver. The validation is part of the openVPN handling

3.4.1.1 Version

For all certificates this field shall contain the integer value 2 to identify the certificate as X.509v3 compliant.

3.4.1.2 Serial Number

The serial number must be a positive integer provided by the CA. The serial number should be a non-sequential numbers to make it more difficult to perform hash collisions attacks or second pre-image attacks.

3.4.1.3 Issuer Signature Algorithm

This field contains the OID of the signature algorithm used by the CA to sign the certificate. In general the signature algorithm is dependent on the signature algorithm used by the CA. It shall be the same as specified within the CA digital signature on this end-entity public-key certificate. If this is not the case, it is an invalid public-key certificate.

The utilized hash algorithm shall use 256 bit keys or stronger. If this is not the case, it is an invalid public-key certificate.

RFC 7427 [9] provides the formal specification for the `sha256WithRSAEncryptionAlgorithm` signature algorithms.

```
sha256WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
  rsadsi(113549) pkcs(1) pkcs-1(1) sha256WithRSAEncryption(11) }
```

RFC 5758[8]provides the formal specification for the `ecdsa-with-SHA256-Algorithm` signature algorithm.

```
ecdsa-with-SHA256   OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
  ansi-x962(10045) signatures(4) ecdsa-with-SHA2(3) 2 }
```

The `ecPublicKey` public-key algorithm requires as a parameter an identification of an elliptic curve. A particular elliptic curve is identified by an object identifier. The following elliptic curves are mandatory to support for this specification.

```
secp256r1          OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
  ansi-x962(10045) curves(3) prime(1) 7 }

brainpoolP256r1    OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
  teletrust(36) algorithm(3) signaturealgorithm(3) ecSign(2) 8 ellipticCurve(1)
  versionOne(1) 7 }
```

Note that the key length of the utilized algorithms should be aligned with the intended validity period of the certificates to be issued. This typically leads to different key length for root CA certificates, subordinate CA certificates or end entity certificates. Possible key length' in the context of this specification depend on the chosen algorithm and may be the following:

- Elliptic curves with key lengths of 256 bits, 384 bits, and 512 bits
- RSA with key lengths 2048 bits, 3072 bits and 4096 bits

(Note that according to BSI TR 02102-1[17], the length of 2048 bit is accepted as minimum till end of 2022).

See also section 3.4.1.7 below.

3.4.1.4 Issuer

This component contains the Distinguished Name (DN) of the issuing CA. The DN value of this component should be globally unique and shall at least be unique within the public-key infrastructure (PKI). The distinguished name should be as simple as possible to reduce size and processing requirements.

3.4.1.5 Validity

The validity period of the certificates depends on the policy of the operator. Typical validity times for different kinds of certificates are:

- Operational end-entity public-key certificate 1-2 years for field devices, 2-3 years for humans.
- Manufacturer installed certificates: depends on the manufacturer and is often not directly determined. Instead of setting the validity to infinity, RFC 5280 [2] recommends the following:
  To indicate that a certificate has no well-defined expiration date, the notAfter SHOULD be assigned the GeneralizedTime value of 99991231235959Z.
- SubCA or RootCA certificate: 10 years.

The CA enters the creation time of the certificate in *notBefore* and adds the validity period to calculate the value of *notAfter*.

Note that the validity period of the CA certificates also influences the CRLs to be kept for a distinct issuing CA certificate (either root CA or subordinate CA).

3.4.1.6 Subject

The Subject carries the unique name or identifier of the device holding the certificate. According to the X.509 standard it is composed of different elements, which are preceded by an attribute. The attributes are listed in [6].

The Subject name shall contain the FQDN of the corresponding device. According to [26] the FQDN is built as

```
fg<FNR>.z<ZNR>.operatordomain
```

3.4.1.7 SubjectPublicKeyInfo

This component contains an OID identifying the signature algorithm of the certificate and the public key type as well as the public key itself. The algorithm OIDs are defined in RFC 3279 [7] and its updates. The following information is taken from RFC 5758 [8].

For the purpose of the OCIT-O Profile 4, the `subjectPublicKeyInfo` component shall be constrained to:

```
SubjectPublicKeyInfo ::= SEQUENCE {
  algorithm         AlgorithmIdentifier{{OCITPublicKeyAlgorithms}},
  subjectPublicKey  PublicKey,
  ... }

OCITPublicKeyAlgorithms ALGORITHM ::= { rsaEncryptionAlgorithm |
                                        ecPublicKey,
                                        ... }
```

where

```
ecPublicKey ALGORITHM ::= {
  PARMS         OCITSupportedCurves
  IDENTIFIED BY id-ecPublicKey }

OCITSupportedCurves OBJECT IDENTIFIER ::= { secp256r1 |
                                            brainpoolPool256r1,
                                            ... }
```

Note that the key length of the utilized algorithms should be aligned with the intended validity period of the certificates to be issued. This typically leads to different key length for root CA certificates, subordinate CA certificates or end entity certificates. Possible key length in the context of this specification depends on the chosen algorithm and may be the following:

- Elliptic curves with key lengths of 256 bits, 384 bits, 512 bits and 521 bits
- RSA with key lengths 2048 bits, 3072 bits and 4096 bits

When **rsaEncryptionAlgorithm** is selected for the public-key algorithm, a minimum key size of 2048 bits shall be used.

(Note that according to BSI TR 02102-1[17], the length of 2048 bit is accepted as minimum till end of 2022).

When **ecPublicKey** is selected for the public-key algorithm, a minimum key size of 256 bits shall be used.

3.4.1.8 Standard Extensions

Additionally to the basic certificate components extensions are used to provide additional information. Standard extensions are defined in RFC 5280 [2] and discussed in the following subsections.

3.4.1.8.1 Authority Key Identifier

The **AuthorityKeyIdentifier** contains the Subject Key Identifier of the CA certificate, which issued the certificate. This extension is used for trust chain validation. This component is mandatory according to RFC5280 [2].

3.4.1.8.2 Subject Key Identifier

The **SubjectKeyIdentifier** holds a 20 byte SHA-1 hash over the public key oft the certificate. It can be used by applications to identify a certain certificate with a known public key more quickly. Its presence is recommended by RFC5280.

The support of this field is mandatory in the context of this specification.

3.4.1.8.3 Key Usage

The **KeyUsage** denotes the elementary usage of a certificate and key. This restriction is important because technically it is often possible to use the key for several purposes (e. g. digital signature and encryption).

This extension is encoded as bit string where every bit represents a certain usage. For the allowed usages the bits are set, the other usages may not be performed with the given certificate:

Bit	Key Usage
0	digitalSignature
1	nonRepudiation
2	keyEncipherment
3	dataEncipherment
4	keyAgreement
5	keyCertSign
6	cRLSign
7	encipherOnly
8	decipherOnly

According to RFC5280 this extension shall be set to critical. An application, which is not able to interpret the key usage, must reject the whole certificate.

The support of this field is mandatory in the context of this specification.

Appendix A.3 provides the values for different certificates expected to be used in an operational environment as certificate profile.

3.4.1.8.4 Basic Constraints

According to RFC 5280 [2], the basic constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths.

3.4.1.8.4.1 CA Constraint

The `cAconstraint` allows explicitly distinguishing CA certificates from end entity certificates. The value CA is set accordingly to true or false. For CA certificates the extension must be critical, for other certificates it is recommended.

The main purpose of this extension is the prevention of attacks against a PKI, where normal end entity certificates are misused for the signing of further, subordinated end-entity certificates.

The support of this field is mandatory in the context of this specification.

3.4.1.8.4.2 Path Length Constraint

According to RFC 5280 [2], the `pathLenConstraint` field is meaningful only if the `cA` Boolean is asserted and the key usage extension, if present, asserts the `keyCertSign` bit. It therefore provides the maximum number of non-self-issued intermediate certificates that may follow this certificate in a valid certification path.

Note: The last certificate in the certification path is not an intermediate certificate, and is not included in this limit.

For OCIT-O Profile 4 this constraint shall be set to 2 for the subCA certificates, allowing a maximum of two intermediate CAs.

Note: The path length depends on the actual operator model and may need to be adjusted.

The support of this field is mandatory in the context of this specification.

3.4.1.8.5 Subject Alternative Name

This component allows including further information to bind the certificate to a certain identity. The Subject Alternative Name mainly offers naming conventions used in the Internet like IP address, FQDN or email address.

The support of this field is optional in the context of this specification.

If the field is supported, it shall contain the following information:

- *Name (CN)*: Contains the device name or identifier. According to OCIT-Profile 3 [12], the name is built as: **`ocit-ZNR-FNR`**
- Additionally, the operator domain shall be provided, resulting in the following: **`ocit-ZNR-FNR.operatordomain`**

It is recommended to limit the attributes in a DN to a subset commonly used in OCIT ODG environments.

3.4.1.8.6 Certificate Revocation List Distribution Point

The Certificate Revocation List Distribution Point (CDP) contains one or more URIs pointing to locations for downloading the certificate revocation list (CRL). The protocols commonly used for fetching a CRL are HTTP or LDAP.

The support of this field is mandatory in the context of this specification.

3.4.1.8.6.1 Authority Information Access

Here one or more URIs point to locations where the issuing CA certificate can be downloaded. Optionally, a URI to an OCSP server can be added.

The support of this field is optional in the context of this specification.

3.4.1.9 Further Extensions

The application of further extensions like for Role-based Access Control or other purposes is for further study.

3.4.1.10 Exchange Format for Enrollment

The certificate shall be DER encoded as PKCS#7 (cf. [11]) in a .p7 file. PKCS#7 has the option to also include the certificate path into the PKCS#7 file.

3.4.2 **Certificate Signing Requests (CSR)**

This section describes required components of a CSR their recommended settings. A CSR shall be encoded as PKCS#10 structure as specified in RFC 2986 [10]. The result of the CSR generation shall be a DER encoded as file with the extension .csr, to be handled by a service technician.

The following ASN.1 structure is taken from RFC 2986 and provides an overview about the context of the CSR:

```
CertificationRequest ::= SEQUENCE {
     certificationRequestInfo CertificationRequestInfo,
     signatureAlgorithm AlgorithmIdentifier{{ SignatureAlgorithms }},
     signature          BIT STRING
}

AlgorithmIdentifier {ALGORITHM:IOSet } ::= SEQUENCE {
     algorithm          ALGORITHM.&id({IOSet}),
     parameters         ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
}

CertificationRequestInfo ::= SEQUENCE {
     version       INTEGER { v1(0) } (v1,...),
     subject       Name,
     subjectPKInfo SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
     attributes    [0] Attributes{{ CRIAttributes }}
}

SubjectPublicKeyInfo { ALGORITHM : IOSet} ::= SEQUENCE {
     algorithm        AlgorithmIdentifier {{IOSet}},
     subjectPublicKey BIT STRING
}

Attributes { ATTRIBUTE:IOSet } ::= SET OF Attribute{{ IOSet }}
```

The following subsections specify the specific settings of the `CertificationRequestInfo` components. For all other, the description within RFC 2986 [10] is referred to.

3.4.2.1 Version

For all CSR this Field contains the integer value 0 and identifies the CSR as compliant to RFC 2986.

3.4.2.2 Subject

The `Subject` carries the unique name or identifier of the device for which the certificate shall be issued. It shall be aligned with the information provided in section 3.4.1.6.

3.4.2.3 SubjectPublicKeyInfo

This component contains information about the public key to be certified. The information identifies the entities public key algorithms and associated parameters and is being provided as an OID. The information to be provided shall match the information in section 3.4.1.7.

3.4.2.4 Attributes

This component may contain additional attributes to be included in the certificate. There are different attribute types defined.

The usage of attributes is left open.

3.4.3 **Certificate Revocation Lists (CRL)**

The utilized CRL format shall be conformant to X.509 (cf. IETF RFC 5280 [2], see also ITU-T X.509 [4] or ISO/IEC 9594-8 [5]). The following ASN.1 structure is taken from RFC 5280 and provides an overview about the context of the CRL:

```
CertificateList  ::=  SEQUENCE  {
     tbsCertList          TBSCertList,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       BIT STRING
}

TBSCertList  ::=  SEQUENCE  {
     version                 Version OPTIONAL,
                                  -- if present, MUST be v2
     signature               AlgorithmIdentifier,
     issuer                  Name,
     thisUpdate              Time,
     nextUpdate              Time OPTIONAL,
     revokedCertificates     SEQUENCE OF SEQUENCE  {
          userCertificate         CertificateSerialNumber,
          revocationDate          Time,
          crlEntryExtensions      Extensions OPTIONAL
                                   -- if present, version MUST be v2
                               }  OPTIONAL,
     crlExtensions           [0]  EXPLICIT Extensions OPTIONAL
                                   -- if present, version MUST be v2
  }
```

The following subsections specify specific settings of the `TBSCertList` components to be used in OCIT-O Profile 4. For all other, the description within RFC 5280 [2] is referred to.

3.4.3.1 Version

For all CRLs this Field contains the integer value 2 and identifies the CRL as X.509v3 compliant.

3.4.3.2 Issuer

Identifies the entity issuing and signing the CRL. This is typically the operator of the CA, but may also be a delegate. The issuer field must contain a distinguished name (DN).

3.4.3.3 thisUpdate

This operator specifies component contains information about the public key to be certified. The information identifies the entities public key algorithms and associated parameters and is being provided as an OID. The information to be provided shall match the information in section 3.4.1.7.

3.5 Resulting Requirements

The following requirements for client and server are also reflected in the conformance statements in Annex A.1.

3.5.1 Requirements for end entity (field device)

Entities claiming conformance with this specification shall support the following:

- Local interface for data exchange for the CSR and also the X.509 certificate with a service technician.

Entities claiming conformance with this specification should support the following:

- Random number generator (RNG):
    - The generation of any random value related to key management should follow ISO/IEC 19790:2012 [13]. Key pair generators shall be responsible for providing statistically adequate random number generators (RNG) and utilizing them appropriately. Note: Guidance can be found in NIST SP 800-90A [14] or the BSI/AIS31 [15] or the ENISA report [16].
    - The interaction with the service technician may be a source of entropy for the RNG.
- An interface for providing a CRL to be checked during the certificate validation. The CRL may be either provided by administration or by fetching the CRL from the CRL distribution point in the certificate.

3.5.2 Requirements for central server

Entities claiming conformance with this specification shall support the following:

- Local interface for data exchange for the CSR and also the X.509 certificate with an end entity.
- An interface for providing a CRL to be checked during the certificate validation. The CRL may be either provided by administration or by fetching the CRL from the CRL distribution point in the certificate.

3.5.3 Requirements for service technician equipment

Entities claiming conformance with this specification shall support the following:

- Local interface for data exchange for the CSR and also the X.509 certificate with an end entity.

Entities claiming conformance with this specification should support the following:

- Ideally an online connectivity to a RA/CA associated with the Operator is available.

3.6 Operational Modes (informative)

The following subsections provide an overview about the different potential operational deployments for a PKI depending on the operator. The final operational model depends on the requirements of the operator and is typically defined by the operator's security policy.

The following operational modes are shortly depicted:

- Root CA operated by a federal agency for communities
- Root CA operated by a federal agency for a country specific Road Construction Office
- Root CA operated by a vendor consortium
- Root CA operated by a control center vendor and installed in the local commune
- Root CA operated and hosted by a control center vendor

3.6.1 Root CA operated by a Federal Agency for Communities

1. Federal Agency (e.g., BSI) operates RootCA for critical infrastructures in the area of transport and traffic → issues sub CA certificates for secure communication for OCIT-O, OCIT-C, and ETSI-G5.
2. Central IT-Department of a Commune operates 1st SubCA→ most likely will manage all community relevant certificates.
3. Operating agency (e.g., public works service) operates 2nd SubCA→ will issue all certificates for control centers, field devices, and RSU from an operator point of view.

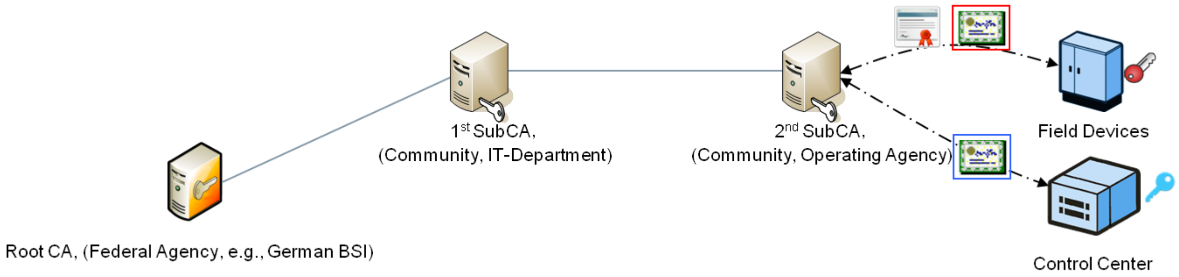
1st SubCA,
(Community, IT-Department)
2nd SubCA,
(Community, Operating Agency)
Field Devices
Control Center
Root CA, (Federal Agency, e.g., German BSI)

Figure 8: RootCA operated by Federal Agency for Communities

3.6.2 Root CA operated by a Federal Agency for a Country specific Road Construction Office

1. Federal Agency (e.g., BSI) operates RootCA for critical infrastructures in the area of transport and traffic → issues sub CA certificates for secure communication for OCIT-O, OCIT-C, and ETSI-G5.
2. Central IT-Department of a Road Construction Office operates 1st SubCA→ most likely will manage all Road Construction Office relevant certificates.
3. Operating office operates 2nd SubCA→ will issue all certificates for control centers, field devices, and RSU from an operator point of view.

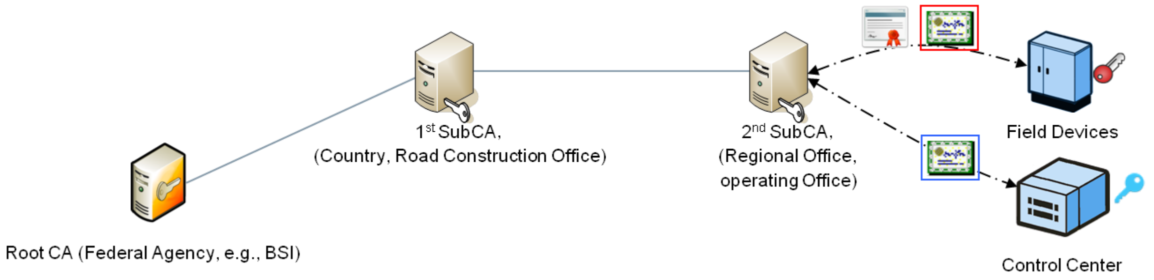

Figure 9: RootCA operated by Federal Agency for Road Construction Office

3.6.3 Root CA operated by a Vendor Consortium

1. Vendor consortium operates RootCA for critical infrastructures in the area of transport and traffic → issues sub CA certificates for secure communication for OCIT-O, OCIT-C, and ETSI-G5.
2. Operating agency (e.g., public works service) operates 1st SubCA → will issue all certificates for control centers, field devices, and RSU from an operator point of view.

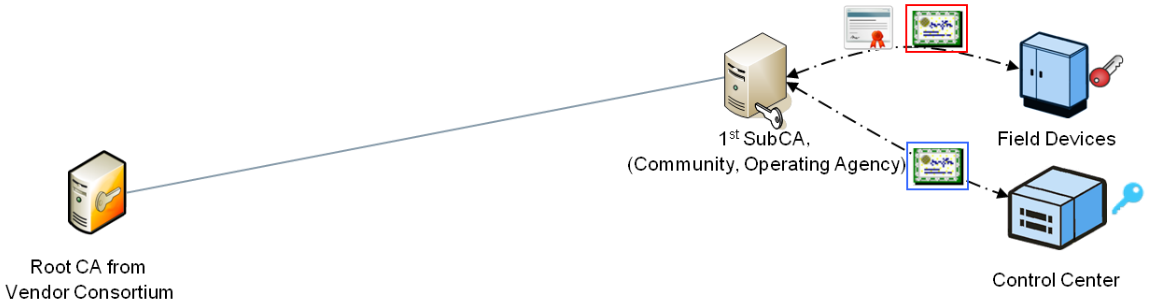

Figure 10: RootCA operated by Vendor Consortium

3.6.4 Root CA operated by a Control Center Vendor and installed in Local Commune

1. Control center vendor operates RootCA for critical infrastructures in the area of transport and traffic → issues sub CA certificates for secure communication for OCIT-O, OCIT-C, and ETSI-G5.
2. Operating agency (e.g., public works service) operates 1st SubCA → will issue all certificates for control centers, field devices, and RSU from an operator point of view.

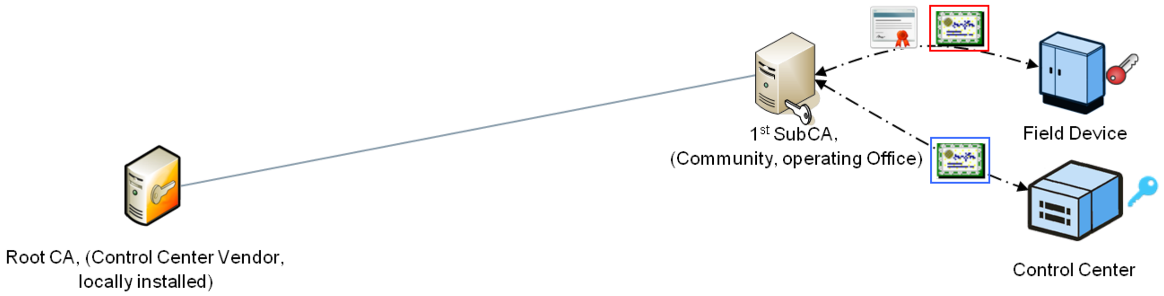
1st SubCA,
(Community, operating Office)
Field Device
Root CA, (Control Center Vendor,
locally installed)
Control Center

Figure 11: RootCA operated by Control Center Vendor

3.6.5 Root CA operated and hosted by a Control Center Vendor

1. Control center vendor operates RootCA for critical infrastructures in the area of transport and traffic → issues sub CA certificates for secure communication for OCIT-O, OCIT-C, and ETSI-G5.
2. Operating agency (e.g., public works service) uses hosted 1st SubCA → will issue all certificates for control centers, field devices, and RSU from an operator point of view.

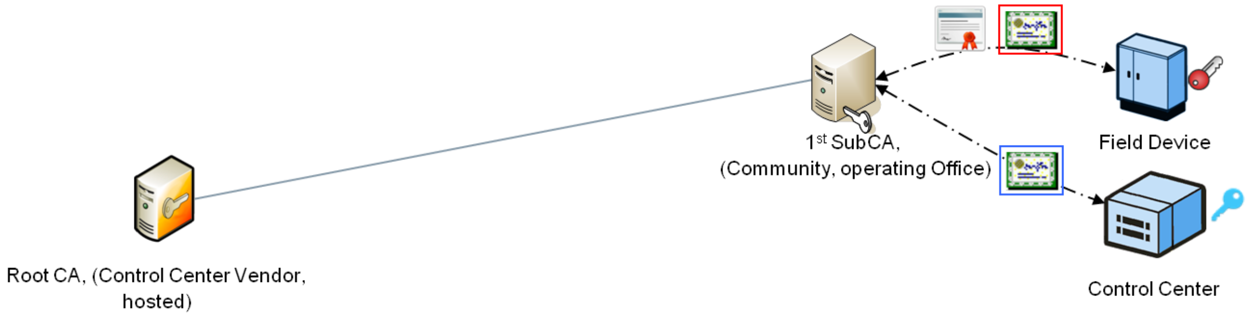
1st SubCA,
(Community, operating Office)
Field Device
Root CA, (Control Center Vendor, hosted)
Control Center

Figure 12: RootCA operated and hosted by Control Center Vendor

A.1 Protocol Implementation Conformance Statement (PICS)

For the definition of PICS, the following definitions apply.

- m: mandatory support. The item shall be implemented.
- c: conditional support. The item shall be implemented if the stated condition exists.
- o: optional support. The implementation may decide to implement the item.
- x: excluded. The implementation shall not implement this item.
- i: out-of-scope. The implementation of the item is not within the scope of this standard.

Item	Description	Client (Field Device)	Server (Central)	Reference
Algorithm Support				
A-1	Support of RSA with key length 2048	c1	m	3.3.1, 3.4.1.3
A-2	Support of RSA with key length 3072	c1	m	3.3.1, 3.4.1.3
A-3	Support of RSA with key length 4096	c1	m	3.3.1, 3.4.1.3
A-4	Support of ECDSA with key length 256	c1	m	3.3.1, 3.4.1.3
A-4.1	Support of secp256r1	c2	m	3.3.1, 3.4.1.3
A-4.2	Support of BrainpoolP256r1	c2	m	3.3.1, 3.4.1.3
A-4	Support of ECDSA with key length 384	o	m	3.3.1, 3.4.1.3
A-4.1	Support of secp384r1	o	m	3.3.1, 3.4.1.3
A-4.2	Support of BrainpoolP384r1	o	m	3.3.1, 3.4.1.3
A-5	Support of ECDSA with key length 512	o	m	3.3.1, 3.4.1.3
A-5.1	Support of secp521r1	o	m	3.3.1, 3.4.1.3
A-5.2	Support of BrainpoolP512r1	o	m	3.3.1, 3.4.1.3
A-6	Support of SHA 256 as hash algorithm	o	m	3.4.1.3
A-7	Support of SHA 384 as hash algorithm	o	m	
A-8	Support of SHA 512 as hash algorithm	o	m	
Control Channel Security Support				
C-1	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	c2	m	2.5.1
C-2	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	c2	m	2.5.1
C-3	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	c4	m	2.5.1
C-4	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	c4	m	2.5.1
C-5	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	c3	m	2.5.1
C-6	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	c3	m	2.5.1
Data Channel Security Support				
D-1	AES-256-GCM	c1	m	2.5.2
D-2	AES-128-GCM	c1	m	2.5.2
D-3	AES-256-CBC	c1	m	2.5.2
D-4	AES-128-CBC	c1	m	2.5.2
D-5	SHA-256	m	m	2.5.2
Management Security Support				
M-1	Support CRL verification functionality	m	m	2.6.3
M-2	CRL verification required during certificate validation	o	m	2.6.3
Protocol Security Support				

Item	Description	Client (Field Device)	Server (Central)	Reference
P-1	Minimum TLS Version requires is 1.2	m	m	**Fehler! Verweisquelle konnte nicht gefunden werden.**
P-1	Minimum OpenVPN Version requires is 2.4	o	m	2.1.2, 2.3
c1: one of these options must be supported. c2: If A-4 is selected, one of these options must be supported. c3: If A-1 or A-2 or A-3 is selected, one of these options must be supported. c4: If A-1 or A-2 or A-3 is selected and support of elliptic curve based algorithms is provided.				

A.2 Security Events

The following table comprises the defined security events in this document.

Relation to	Description	Reference
VPN tunnel states	warning: tunnel state error	2.2.3, 2.3.4
Certificate availability	warning: certificate unavailable	2.6.1,
	warning: CA not found	2.6.1,
Certificate Policy	incident: Certificate parameter wrong	2.6.3
Certificate Revocation	warning: CRL not accessible	2.6.3
	warning: CRL expired	2.6.3
	incident: revoked certificate	2.6.3
Certificate Expiry	warning: expired certificate	2.6.4
Protocol handshake errors	incident: session renegotiation interval expired	2.7
	incident: unsecure communication	2.7

A.3 Certificate Profile

The following certificate profiles are provided here as helper. The markers used are the following:

x Required, must be sent and processed

(x) Optional, can be sent. If included, receiver must be able to process the field

- Must not be present, sender shall not send this item

c This extension is critical, see IETF RFC 5280. If a implementation recognizes that a “critical” extension is present, but the implementation cannot interpret the extension, the implementation has to reject the certificate.

nc This extension is non-critical, see IETF RFC 5280. If a implementation recognizes that a “non-critical” extension is present, but the implementation cannot interpret the extension, the extension can be ignored. Therefore, all optional fields should also be “non-critical”.
Quote from RFC 5280: “A certificate-using system SHALL reject the certificate if it encounters a critical extension it does not recognize or a critical extension that contains information that it cannot process. A non-critical extension MAY be ignored if it is not recognized, but SHALL be processed if it is recognized."

1 Bit shall be sent with value 1

0 Bit shall be sent with value 0

0/1 Bit can be sent with value 0 or 1, the sender is free to choose the value

A.3.1 OCIT Operator System

OCIT-O Profile 4 Certificate Profiles User Group			Cluster:		System Operator (SO)			
			Name:	DEFAULT	Root CA	Sub-CA 1	Sub-CA 2	Entity Certificate
			Typ:	Root/Sub/Leaf	Root	Sub	Sub	Leaf
tbsCertificate	Version			2 (X.509v3)	2 (X.509v3)	2 (X.509v3)	2 (X.509v3)	2 (X.509v3)
	SerialNumber			Integer	Integer	Integer	Integer	Integer
	Signature			rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256
Issuer	Country			(x)	(x)	(x)	(x)	(x)
	Organization			x	x	x	x	x
	Organization Unit			(x)	(x)	(x)	(x)	(x)
	Common Name			x	x	x	x	x
	Domain Component			(x)	(x)	(x)	(x)	(x)
Validity					[SO policy]	[SO policy]	[SO policy]	[SO policy]
Subject	Country			(x)	(x)	(x)	(x)	-
	Organization			x	x	x	x	x
	Organization Unit			(x)	(x)	(x)	(x)	(x)
	Common Name			x	x	x	x	x
	Domain Component			(x)	(x)	(x)	(x)	(x)
SubjectPublicKeyInfo	Public Key			x	x	x	x	x
	Cryptographic Algorithm			id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey
	Parameters			ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)
Extensions	AuthorityKeyIdentifier			x / nc	x / nc	x / nc	x / nc	x / nc
	SubjectKeyIdentifier			(x) / nc	(x) / nc	(x) / nc	(x) / nc	(x) / nc
	KeyUsage			c	c	c	c	c
		digitalSignature		0/1	0/1	0/1	0/1	1
		nonRepudiation (contentCommitment)		0/1	0/1	0/1	0/1	1
		keyEncipherment		0/1	0/1	0/1	0/1	1
		dataEncipherment		0	0	0	0	0
		keyAgreement		0/1	0/1	0/1	0/1	1
		keyCertSign		1	1	1	1	0
		cRLSign		1	1	1	1	0
		encipherOnly		0	0	0	0	0
		decipherOnly		0	0	0	0	0
	ExtendedKeyUsage			-	-	-	-	-
	BasicConstraints			c	c	c	c	c
		CA		TRUE	TRUE	TRUE	TRUE	FALSE
		PathLength		-	-	1	2	-
	CRLDistributionPoints			(x) / nc	(x) / nc	(x) / nc	(x) / nc	(x) / nc
	Authority Information Access (OCSP)			(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder
Signature Value	Cryptographic Algorithm Signature Value			rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String

A.3.2 OEM Certificates

OCIT-O Profile 4 Certificate Profiles User Group			Cluster: Name: Typ:	DEFAULT Root/Sub/Leaf	Original Equipment Manufacturer (OEM)		
					OEM Root CA Root	OEM Sub-CA 1 Sub	OEM Prov Certificate Leaf
tbsCertificate	Version			2 (X.509v3)	2 (X.509v3)	2 (X.509v3)	2 (X.509v3)
	SerialNumber			Integer	Integer	Integer	Integer
	Signature			rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256	rsa-with-SHA256, ecdsa-with-SHA256
Issuer	Country			(x)	(x)	(x)	(x)
	Organization			x	x	x	x
	Organization Unit			(x)	(x)	(x)	(x)
	Common Name			x	x	x	x
	Domain Component			(x)	(x)	(x)	(x)
Validity					[OEM policy]	[OEM policy]	[OEM policy]
Subject	Country			(x)	(x)	(x)	-
	Organization			x	x	x	x
	Organization Unit			(x)	(x)	(x)	(x)
	Common Name			x	x	x	x
	Domain Component			(x)	(x)	(x)	(x)
SubjectPublicKeyInfo	Public Key			x	x	x	x
	Cryptographic Algorithm			id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey	id-rsaPublicKey, id-ecPublicKey
	Parameters			ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)	ECParameters (namedCurve secp256r1)
Extensions	AuthorityKeyIdentifier			x / nc	x / nc	x / nc	x / nc
	SubjectKeyIdentifier			(x) / nc	(x) / nc	(x) / nc	(x) / nc
	KeyUsage			c	c	c	c
		digitalSignature		0/1	0/1	0/1	1
		nonRepudiation (contentCommitment)		0/1	0/1	0/1	1
		keyEncipherment		0/1	0/1	0/1	1
		dataEncipherment		0	0	0	0
		keyAgreement		0/1	0/1	0/1	1
		keyCertSign		1	1	1	0
		cRLSign		1	1	1	0
		encipherOnly		0	0	0	0
		decipherOnly		0	0	0	0
	ExtendedKeyUsage			-	-	-	-
	BasicConstraints			c	c	c	c
		CA		TRUE	TRUE	TRUE	FALSE
		PathLength		-	-	1	-
	CRLDistributionPoints			(x) / nc	(x) / nc	(x) / nc	(x) / nc
	Authority Information Access (OCSP)			(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder	(x) / nc id-ad-ocsp / location of the OCSP responder
Signature Value	Cryptographic Algorithm Signature Value			rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String	rsa-with-SHA256, ecdsa-with-SHA256 Octet-String

Glossary

AES	Advanced Encryption Standard
BTPPL	Basis Transport Packet Protocol Layer of the OCIT-O interface
CA	Certificate Authority
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DNS	Domain Name System
ECDSA	Elliptic Curve Digital Signature Algorithm
FNr	Number of a field device belonging to a control center. All devices, which can be controlled by the same control center, must be uniquely defined in the realm of the control center.
IED	Intelligent Electronic Device
IP	Internet Protocol (Version 4, if not otherwise noted)
ISO / OSI	ISO/OSI-Basic Reference Model (DIN-ISO 7498 v.1982, X.200 v. 1994) ISO: International Organization for Standardization OSI: Open Systems Interconnection
GPRS	General Packet Radio Services
LAN	Local Area Network
LTE	Long Term Evolution
OpenVPN	OpenVPN allows the establishment of Virtual Private Network (VPN) (cf. open-vpn.net)
RA	Registration Authority
RFC	Request for Comment (Protocol specifications from the IETF)
RootCA	Root Certification Authority
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman, asymmetric cryptographic algorithm
Security Event	A security event generated by an instance is typically logged by the instance and may lead to derived actions. Security events may be transmitted to other system entities using either OCIT inherit means or protocols like syslog or SNMP.
SubCA	Subordinate Certification Authority
SHA	Secure Hash Algorithm
TCP	Transmission Control Protocol One of the internet protocols. Connection-oriented transport protocol on layer 4 of the ISO/OSI reference model.
VPN	Virtual Private Net
UDP	User Datagram Protocol One of the internet protocols. Connection-less transport protocol on layer 4 of the ISO/OSI reference model.
ZNr	Number of the control center. Every control center of an operator must have a unique number.

References

[1] RFC 2119: *Key words for use in RFCs to Indicate Requirement Levels*, S. Bradner, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[2] RFC 5280: *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, May 2008, http://www.ietf.org/rfc/rfc5280.txt

[3] FIPS 186-4: Digital Signature Standard (DSS), NIST, 2014, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[4] ITU-T X.509: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks, 2014, http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11735

[5] ISO/IEC 9594-8: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks, 2014 http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=64854

[6] X.500 Attribute types, http://www.alvestrand.no/objectid/2.5.4.html

[7] RFC 3279: *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, W. Polk, R. Housley, L. Bassham, April 2002 https://tools.ietf.org/html/rfc3279

[8] RFC 5758: *Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA*, Q. Dang, S. Santesson, K. Moriarty, D. Brown, T. Polk, January 2010 https://tools.ietf.org/html/rfc5758

[9] RFC 7427: *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*, T. Kivinen, J. Snyder, January 2015 https://tools.ietf.org/html/rfc7427

[10] RFC 2986: *PKCS#10: Certification Request Syntax Specification Version 1.7*, M. Nystrom, B. Kaliski, November 2000, https://tools.ietf.org/html/rfc2986

[11] RFC 2315: *PKCS #7: Cryptographic Message Syntax Version 1.5*, B. Kaliski, March 2003, https://tools.ietf.org/html/rfc2315

[12] OCIT-O Profil 3: *Ethernet mit DHCP*, 2012 http://www.ocit.org/specs_public/Profil_3/OCIT-O-Profil_3_V1.0_A02.pdf

[13] IEC 19790:2012: *Information technology - Security techniques - Security requirements for cryptographic modules*

[14] NIST SP 800-90 A: *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, Elaine Barker and John Kelsey, January 2012, http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

[15] BSI/AIS31: *A proposal for: Functionality classes for random number generators*, September 2011 https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Zertifierung/Interpretation/AIS31_Functionality_classes_for_random_number_generators.pdf

[16] ENISA: *Algorithms, key size and parameters report – 2014*, 2014 https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014

[17] BSI TR02102-1: *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, Version 02/2018, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf

[18] BSI TR02102-2: *Verwendung von Transport Layer Security (TLS)*, Version 01/2018, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.pdf

[19] NIST SP 800-52rev.1: *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*, 04/2014
http://dx.doi.org/10.6028/NIST.SP.800-52r1

[20] *SEC2: Recommended Elliptic Curve Domain Parameters*, 01/2010,
http://www.secg.org/sec2-v2.pdf

[21] RFC 5639: *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*, M. Lochter, J. Merkle, 03/2010
https://tools.ietf.org/html/rfc5639

[22] RFC 7027: *Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS)*, M. Lochter, J. Merkle, 10/2013
https://tools.ietf.org/html/rfc7027

[23] OpenSSL, www.openssl.org

[24] OpenVPN, openvpn.net/

[25] RFC 5246: *The Transport Layer Security (TLS) Protocol Version 1.2*, T. Dierks, E. Rescorla, 08/2008
https://tools.ietf.org/html/rfc5246

[26] OCIT FAQ, http://www.ocit.org/FAQ.htm

[27] OCIT-Outstations Basisfunktionen für Feldgeräte, Version 2018, https://www.ocit.org/media/ocit-o_basis_v3.0_a01.pdf