Annex Profiles for Certification Practice Statement certSIGN ROOT CA

Version 1.45
Date: 15 January 2026

Address: 29 A Tudor Vladimirescu Avenue,
AFI Tech Park 1, Bucharest, Romania
Phone: 004-021-31.19.901

Web: www.certsign.ro

Document History

Version	Effective Date[1]	Reason	The person who made the change
1.40	July 2023	First version published	PKI Policies Manager
1.41	August 2023	Remove emailProtection	PKI Policies Manager
1.41a	January 2024	Annual review	PKI Policies Manager
1.42	18 April 2024	Add cross-certificate	PKI Policies Manager
1.43	15 January 2025	Annual review	PKI Policies Manager
1.44	15 April 2025	Updates in footer	PKI Policies Manager
1.45	15 January 2026	Annual review	PKI Policies Manager

This document was approved by:

Version	Name	Date
1.40	Policies and Procedures Management Body	July 2023
1.41a	Policies and Procedures Management Body	January 2024
1.43	Policies and Procedures Management Body	January 2025
1.44	Policies and Procedures Management Body	April 2025
1.45	Policies and Procedures Management Body	January 2026

Content

[1] *Effective date is the las day of the month*

7 Certificate, CRL and OCSP profile

Certificate profiles and Certificate Revocation List (CRL) profile comply with the format described in ITU-T X.509 v.3 standard, while the profile of OCSP complies with the requirements of RFC 6960. Information stated below describes the meaning of respective certificate fields, CRL and OCSP, applied standard and private extensions used by CERTSIGN.

7.1 Certificate profile

Following the X.509 v.3 standard, a certificate is the sequence of the following fields: the first the body of certificate (**tbsCertificate**), information about algorithm used for certificate signing (**signatureAlgorithm**), and an electronic signature of the Certification Authority (**signatureValue**).

7.1.1 Version number(s)

The contents of a certificate include values of **basic fields** and **extensions** (standard, described by norms, and private, defined by the issuing authority).

Extensions defined in a certificate according to norms allow assignation of additional attributes to the Subscriber and his/her/its public key and simplify management of hierarchical certificate structure. Certificates issued in accordance with X.509 v.3 standard allow definition of proprietary extensions, unique for a given implementation. Basic fields

certSIGN supports the following basic fields:

- **Version**: third version (X.509 v.3) of certificate format,
- **SerialNumber**: certificate serial number, unique within Certification Authority domain,
- **signatureAlgorithm**: identifier of the algorithm applied by a issuing Certification Authority,
- **Issuer**: distinguished name (DN) of a Certification Authority,
- **Validity**: validity period, described by the beginning date (**notBefore**) and the ending date (**notAfter**) of the certificate,
- **Subject**: distinguished name (DN) of the Subscriber that is the subject of the certificate,
- **SubjectPublicKeyInfo**: value of a public key along with the identifier of the used cryptographic algorithm associated with the key.

In certificates issued by certSIGN values of the above fields are set in accordance with rules described in Table 7.1.

Field name	Value or value's constraint	
Version	Version 3	
Serial Number	Unique value for all certificate issued by Certification Authorities within certSIGN. In this field will be introduced a random value of 8 bytes. A hardware cryptographic module will be used for generating this value.	
SIGNature Algorithm	sha256WithRSAEncryption (OID: 1.2.840.113549.1.1.11)	
Issuer (Distinguished Name)	Name (CN) =	certSIGN {CA Class {1,2,3,4}}
	Department (OU)=	certSIGN {CA Class {1,2,3,4}}
	Organization (O) =	certSIGN
	Country (C) =	RO
Not before (validity period beginning date)	Universal Time Coordinated based. certSIGN owns a satellite clock controlled by Atomic Frequency Standard.	
Not after (validity period end date)	Universal Time Coordinated based. certSIGN owns a satellite clock controlled by Atomic Frequency Standard.	
Subject (Distinguished Name)	Distinguished names comply with the X.501 requirements. Values of all attributes of these fields are optional and their signification is described below.	
Subject Public Key Info	Encoded in accordance with RFC 3280, may contain information about RSA, DSA or ECDSA public keys (key identifier, key size in bits and value of the public key); RSA key size is presented in Chapter 6.1.5.	
Signature	Certificate signature, generated and encoded in accordance with the requirements described in RFC 3280.	

Table 7.1. Profile of the basic fields of certificates

Profiles of all certificate (Subject fields):

Subordinate CA Certificate Profile
All subject names are encoded according to the specifications in CABF BR section 7.1.4 and contain AttributeTypes according to #7.1.2.10.2 "CA Certificate Naming".

Cross-Certified Subordinate CA Certificate Profile
This Certificate Profile is used when issuing a CA Certificate using the same Subject Name and Subject Public Key Information as one or more existing Subordinate CA Certificate.

7.1.2 Certificate extensions

Function of every extension is defined by the standard value of the corresponding object identifier (**OBJECT IDENTIFIER**). Extension, depending of the choice of issuing authority, may be **critical** or **non-critical**. If an extension is defined as **critical**, the application

supporting certificate usage must reject every certificate containing an unrecognized critical extension. On the other hand, extensions defined as **non-critical** may be omitted.

certSIGN supports the following fields of standard extensions:

- **AuthorityKeyIdentifier**: identifier of a Certification Authority public key certificate associated with a private key, used for signing issued certificates – **this extension in not critical**,
- **SubjectKeyIdentifier** – subject key identifier – **this extension in not critical**,
- **KeyUsage**: allowed key usage – **this extension is critical**. This extension describes the usage of the key, e.g. key for data encryption, key for data exchange, key for electronic signature, etc:

  digitalsignature (0) – key for electronic signature creation

  nonRepudiation (1) – key associated with the non-repudiation services

  keyEncipherment (2) – key for key exchange

  dataEncipherment (3) – key for data encryption

  keyAgreement (4) – key for key agreement

  keycertsign (5) – key for certificate signing

  CRLsign (6) – key for CRL signing

  encipherOnly (7) – key only for encryption

  decipherOnly (8) – key only for decryption

- **ExtKeyUsage**: defines the constraints related to the key usage – **this extension is not critical**. This field defines one or more areas, in addition to standard key usage, defined by **keyUsage** field, of the possible usage of a certificate. This field should be interpreted as constraint of allowed key usage purpose defined in field **keyUsage**. certSIGN issues certificates which may contain one of the following value or combination of such values in ExtKeyUsage field:

  serverAuth – authentication of TLS Web servers; **keyUsage** field bits are set for: digitalSignature, keyEncipherment or keyAgreement

  clientAuth – authentication of TLS Web clients; **keyUsage** field bits are set for: digitalSignatureand/or keyAgreement

  codesigning – signature of executable codes; **keyUsage** field bits set for digitalSignature

emailProtection – E-mail protection; keyUsage field bits set for: digitalSignature, nonRepudiation and/or (keyEncipherment or keyAgreement)

ipsecEndSystem – IPSEC protocol protection,

ipsecTunnel – IPSEC protocol Tunnelling,

ipsecUser – IP protocol protection in user application,

timeStamping – binding of the digest value with the time provided by previously accepted trusted time source; **keyUsage** field bits are set for:digitalSignature, nonRepudiation.

OCSPsigning – assigns the right to issue certificate status confirmations on behalf of CA; keyUsage field bits are set for: digitalSignature, nonRepudiation

dvcs – issuance of confirmation by a notary authority, on the basis of DVCS protocol; keyUsage field bits are set for: digitalSignature, nonRepudiation, keyCertSign, cRLSign

EncryptedFileSystem – allows the usage of the certificate to encrypt the file system (EFS); it is a mandatory request from certain applications (i.e. EFS);

SmartCardLogon – allows the usage of the certificate for „smart-card logon" operation – authentication in the operating system, based on the digital certificate;

- **Certificate Policies** – the extension indicates the policy (policies) based on a Certification Authority will issue certificates or the policy (policies) based on which a Certification Authority issued a certificate. The extension is a **PolicyInformation** list–information (identifier, electronic address) about an applied certification policy. **This extension is not critical**.

Certification Policy Name	Policy identifier
certSIGN Class 1	**{certSIGN}[2] .{id-policy}[2]. {id-cp}[3].{id-Class-1}[4] =**1.3.6.1.4.1.25017.1.1.1
certSIGN Class 2	**{certSIGN} id-policy(1) id-cp(1)id-Class-2(2)=** 1.3.6.1.4.1.25017.1.1.2
certSIGN Class 3	**{certSIGN} id-policy(1) id-cp(1)id-Class-3(3) =**1.3.6.1.4.1.25017.1.1.3
certSIGN Class 4	**{certSIGN} id-policy(1) id-cp(1)id-Class-4(4) =**1.3.6.1.4.1.25017.1.1.4

Table 7.1.2.Policies identifiers and their names

[2] {certSIGN}=1.3.6.1.4.1.25017= iso(1). identified-organization(3). dod(6). internet(1). private(4). enterprise(1). certSIGN's IANNA assigned number (20715); [2] {id-policy}=1; [3] {id-cp}=1;[4] {Class-1}=1

Certificates issued by Certification Authorities include also qualifiers, recommended by the RFC 3280.

- **PolicyMapping**: policy mapping – **this field is not critical**; this field contains one or more pairs of OID, defining equivalency of the certificate issuer policy with the certificate subject policy,
- **SubjectAlternativeName:** alternative name of the subject – this field is not critical,
- **BasicConstraints**: basic constrains – defines the certificate type (CA or end entity certificate), as well as the maximum accepted length for the certificate chain – **this field is critical;**
- **CRL DistributionPoints**: point of distribution of Certificate Revocation List – **this field in not critical**; the extension defines network addresses hosting the current CLR of the issuer Authority of the respective certificate,
- **AuthorityInfoAccessSyntax**: access to Certification Authority information – **this field is not critical**; the field indicates the method of information and service provision by the issuer of the certificate,
- **OCSPNoCheck**: if it is included in a OCSP responder certificate, the clients who receive OCSP responses signed with a private key associated to the certificate may trust the certificate status during its availability period; this extension **is not critical** and it is defined by the RFC 6960 standard.
- **NetscapeCertType**: this extension limits the certificate usage only to certain applications defined by the extension's value. If it is not present, the certificate may be used for any application except the ObjectSigning applications. This extension **is not critical**, and its value may be one of the following combinations:

  SSLClient (bit 0) – certificate may be used to authenticate a SSL client

  SSLServer (bit 1) – certificate may be used to authenticate a SSL server

  S/MIME (bit 2) – certificate may be used by clients of S/MIME secured mail

  ObjectSigning (bit 3) – certificate may be used to sign objects such as Java applets or plug-ins

  SSL CA (bit 5) – certificate may be used to issue certificates used for SSL

  S/MIME CA (bit 6) – certificate may be used to issue certificates used for S/MIME

  ObjectSigning CA (bit 7) – certificate may be used for issuing certificates used for ObjectSigning

  Observation: for the value of NetscapeCertType extension, bit 4 is not yet defined as being reserved for a future usage

Certificates issued by certSIGN may contain various combinations of extensions:

7.1.2.1 Certification Authorities certificates

The certificates profiles extensions are according to CABF BR # 7.1.2 "Certificate Content and Extensions".
The ***AuthorityInfoAccess*** contain one or more AccessDescriptions. Each AccessDescription only contains a permitted accessMethod, and each accessLocation is encoded as the specified GeneralName type.

Authority Key Identifier extension have only the ***keyIdentifier*** field, identical to the subjectKeyIdentifier field of the Issuing CA.
CA Certificate ***Basic Constraints***: cA set TRUE; pathLenConstraint set to 0 or NULL.
Certificate Policies extension contains at least one PolicyInformation and it contain exactly one Reserved Certificate Policy Identifier – for CA certificates issued after 2023.
The ***CRL Distribution Points*** extension contains at least one DistributionPoint, of type uniformResourceIdentifier, and the scheme of each is "http". The first GeneralName contains the HTTP URL of the Issuing CA's CRL service for the CA certificate.
certSIGN CA generates a ***subjectKeyIdentifier*** that is unique within the scope of all Certificates it has issued for each unique public key.
For CA certificates issued after 2023 - the CA Certificate ***Extended Key Usage*** contains id-kp-serverAuth key, and optionally id-kp-clientAuth.

A certificate issued for Certification Authorities may contain extension from Table 7.1.3 and 7.1.5.

Extension	Value or Value constraint	Extension status
Basic Constraints	Subject type=CA, Path length constraint=none	Critical
Key Usage	digitalSignature (bit 0), nonRepudiation (bit 1), keyCertSign (bit 5), cRLSign (bit 6)	Critical

Table 7.1.3. Extensions of certSIGN Root CA certificate

Extension	Value or Value constraint	Extension status
Basic Constraints	Subject type=CA, Path length constraint=0	Critical
Key Usage	digitalSignature (bit 0), nonRepudiation (bit 1), keyCertSign (bit 5), cRLSign (bit 6)	Critical
CRL Distribution Points	http://crl.certsign.ro/root.crl	Non-critical
Certificate Policies	Policies: 1.3.6.1.4.1.25017.1.1.3 CPS: http://www.certsign.ro/repository	Non-critical
Authority Info Access	OCSP: http://ocsp.certsign.ro ISSUER: http://crl.certsign.ro/root.crt	Non-critical

Table 7.1.5 Extensions of the certificates for Intermediate Authoritie (Classes 2-4) G2

7.1.2.2 Cross-certification and non-repudiation certificates

Cross-certification and non-repudiation certificates may contain extensions specified in Table 7.1.13, 7.1.14, 7.1.15 and 7.1.16.

Extension	Value or Value constraint	Extension status
Basic Constraints	Subject type=CA, Path length constraint={none,1,2,...}	Critical
Key Usage	digitalSignature (bit 0), nonRepudiation (bit 1), keyCertSign (bit 5) cRLSign (bit 6)	Critical
CRL Distribution Points	URI: http://crl.certSIGN.ro/class4.crl http://crl.certsign.ro/class4g2.crl ldap://ldap.certSIGN.ro/C=RO,O=certSIGN,OU=certSIGN Class 4?certificateRevocationList;binary	Non-critical
Authority Info Access	OCSP: http://ocsp.certSIGN.ro	Non-critical
Certificate Policies	Politicile:1.3.6.1.4.1.25017.1.1.4 CPS: http://www.certSIGN.ro/repository	Non-critical

Table 7.1.13. Non-repudiation certificates extensions

OCSP Responder Certificate Profile Extensions
Authority Key Identifier extension have only the ***keyIdentifier*** field, identical to the subjectKeyIdentifier field of the Issuing CA.
OCSP Responder Extended Key Usage is only OCSP Signing (1.3.6.1.5.5.7.3.9).certSIGN includes the ***id-pkix-ocsp-nocheck*** extension (OID: 1.3.6.1.5.5.7.48.1.5).
This extension has an extnValue OCTET STRING which is exactly the hex-encoded bytes 0500, the encoded representation of the ASN.1 NULL value, as specified in RFC 6960, Section 4.2.2.2.1.
OCSP Responder ***Key Usage*** is only digitalSignature.
subjectAltName, authorityInformationAccess, certificatePolicies, crlDistributionPoints are not set as extensions for OCSP certificates issued after 2023.

Extension	Value or Value constraint	Extension status
Key Usage	digitalSignature (bit 0), ~~nonRepudiation (bit 1)~~	Critical
Extended Key Usage	OCSPSigning	Non-critical
OCSPNoCheck	-	Non-critical
Certificate Policies	Politicile:1.3.6.1.4.1.25017.1.1.{2.3.4}.1 CPS: http://www.certSIGN.ro/repository	Non-critical

Table 7.1.14 OCSP Authority certificates' extensions

Extension	Value or Value constraint	Extension status
Basic Constraints	Subject type=CA, Path length constraint=0	Critical
Key Usage	keyCertSign (bit 5) cRLSign (bit 6)	Critical
CRL Distribution Points	URI: http://crl.certsign.ro/root.crl	Non-critical
Authority Info Access	OCSP: http://ocsp.certsign.ro CRT: http://www.certsign.ro/certcrl/root.crt	Non-critical
Certificate Policies	Politicile:1.3.6.1.4.1.25017.1.1.3 CPS: http://www.certSIGN.ro/repository	Non-critical
Extended Key Usage	Server Authentication (1.3.6.1.5.5.7.3.1) Client Authentication (1.3.6.1.5.5.7.3.2)	Non-critical

Table 7.1.16. Cross-certification for certSIGN Web CA extensions

Besides the extensions mentioned above, upon the client's request the certificates may include also particular extensions, under the conditions settled on concluding the contract.

7.1.3 Algorithm object identifiers

SubjectPublicKeyInfo

The SubjectPublicKeyInfo field indicate an RSA key using the rsaEncryption (OID: 1.2.840.113549.1.1.1) algorithm identifier, with an explicit NULL parameter.
The AlgorithmIdentifier for RSA keys is byte-for-byte identical with the following hex-encoded bytes: 300d06092a864886f70d0101010500.
For ECDSA, the identifiers and encodings specified in #7.1.3.1.2 from CABF BR will be used.

Signature AlgorithmIdentifier
All TLS objects signed by a certSIGN CA Private Key conform to the CABF BR #7.1.3.2, RSA or ECDSA requirements on the use of the AlgorithmIdentifier or AlgorithmIdentifier-derived type in the context of signatures. In the case of certSIGN, the algorithm used is sha256WithRSAEncryption (OID: 1.2.840.113549.1.1.11).

7.1.4 Name forms

Name Encoding
The contents of the fields must meet the requirements in section 3.1 in CPS and the latest published version of CAB Forum BR.
Issuer Name for all possible certification paths is byte-for-byte identical with Subject Name of the Issuer certificate. Subject attributes do not contain only metadata such as '.', '-', and ' ' (i.e. space) to indicate that the value is absent, incomplete, or not applicable.
For every valid Certification Path (as defined by RFC 5280, Section 6):
• For each Certificate in the Certification Path, the encoded content of the Issuer Distinguished Name field of a Certificate is byte-for-byte identical with the encoded form of the Subject Distinguished Name field of the Issuing CA certificate.
• For each TLS CA Certificate in the Certification Path, the encoded content of the Subject Distinguished Name field of a Certificate is byte-for-byte identical among all Certificates whose Subject Distinguished Names can be compared as equal according to RFC 5280, Section 7.1, and including expired and revoked Certificates.

TLS Subject Attribute Encoding
The attributes in the Certificate subject field will be encoded and positioned according to Table 77: "*Encoding and Order Requirements for Selected Attributes*" from CBAF BR section 7.1.4.2 Subject Attribute Encoding.

Subscriber TLS Certificate Common Name Attribute
This attribute contains exactly one entry that is one of the values contained in the Certificate's subjectAltName extension.
If the value is a Fully-Qualified Domain Name or Wildcard Domain Name, then the value is encoded as a character-for-character copy of the dNSName entry value from the subjectAltName extension. Specifically, all Domain Labels of the Fully-Qualified Domain Name or FQDN portion of the Wildcard Domain Name must be encoded as LDH Labels, and P-Labels will NOT be converted to their Unicode representation.

7.1.5 Name constraints

Not applicable.

7.1.6 Certificate policy object identifier

Certificates policy object identifiers used at certSIGN Root CA are described in Table 7.1.2

7.1.7 Usage of Policy Constraints extension

Not applicable.

7.1.8 Policy qualifiers syntax and semantics

CERTSIGN issue certificates with a policy qualifier within the Certificate Policies extension. This extension contains a CPS pointer qualifier that points to the CPS.

7.1.9 Processing semantics for the critical Certificate Policies extension

Not applicable.

7.2 CRL profile

Certificate Revocation List (CRL) consists of three fields. The first field (**tbscertList**) contains information about revoked certificates, the second and the third field -**signatureAlgorithm** and **signatureValue** contain information about respectively: the identifier of the algorithm used for list signing, and electronic signature of the Certification Authority.
The field of **tbscertList** is the sequence of mandatory and optional fields. Mandatory fields identify CRL issuer, while optional fields contain information about revoked certificates and CRL extensions.
The following fields are the contents of mandatory and optional fields of CRL:

- **Version**: CRL format version,
- **signature**: contains identifier of the algorithm used by a Certification authority to sign CRL; certSIGN authorities sign CRLs by means of **sha256WithRSAEncryption** algorithm,
- **Issuer**: name of the Certification Authority issuing CRL; every authority of certSIGN issues its own Certificate Revocation List; this requirement applies to the following authorities: **certSIGN SSL DV CA Class 3 G2**
- **ThisUpdate**: CRL publication date,
- **NextUpdate**: announcement of the date of the next CRL publication; if the field is present, its value describes the maximum date for CRL update,
- **Revokedcertificates**: the list of revoked certificates (the field is empty in the case of lack of revoked certificates); the information consists of three sub-fields:
  usercertificates – serial number of a revoked certificate;
  revocationDate – date of the certificate revocation;
  crlEntryExtensions – contains additional information about revoked certificates – optional.
- **crlExtensions**: extended information about Certificate Revocation List (optional field). Among numerous extensions, the most important are the following ones: **AuthorityKeyIdentifier** (see also Chapter 7.1.1.2) allowing identification of a public key corresponding to a private key used for list signing, and **cRLNumber**, containing monotonically increased serial number of the lists issued by a Certification Authority (by means of this extension, a Subscriber is able to define when a specific CRL replaced another list).

7.2.1 Version numbers (s)

All CRLs issued by CERTSIGN are X.509 version 2.

7.2.2 CRL and CRL entry extensions

CRLNumber extension contains an INTEGER greater than or equal to zero (0) and less than 2^{159}, and convey a strictly increasing sequence.

serialNumber is byte-for-byte identical to the ***serialNumber*** contained in the revoked Certificate.

revocationDate is the date and time revocation occurred.
The CA updates the revocation date in a CRL entry when it is determined that the private key of the Certificate was compromised prior to the revocation date that is indicated in the CRL entry for that Certificate. Backdating the revocationDate field is an exception to best practice described in RFC 5280 (Section 5.3.2); the revocationDate field support TLS implementations that process the revocationDate field as the date when the Certificate is first considered to be compromised.

Function and meaning of extensions are the same as for certificate extensions (see Chapter 7.1.2). CRL entry extensions (**crlEntryExtensions**) supported by certSIGN contain the following fields:

ReasonCode: code of the reason for revocation. This field is **non-critical**, allowing determination of the certificate revocation reason. The following reasons of certificate revocation are allowed, for non-SSL/TLS certificates:

- **unspecified** – not specified**keyCompromise** – key compromising;
- **cACompromise** – Certification Authority key compromising;
- **affiliationChanged** – Subscriber's data modification;
- **superseded** – certificate renewal;
- **cessationOfOperation** – cessation of certificate usage;
- privilegeWithdrawn – pierderea drepturilor;
- **certificateHold** – certificate suspension;
- **removeFromCRL** – certificate removal from CRL;

The ReasonCode values **unspecified** – not specified and **certificateHold** – certificate suspension are NOT permitted on revoking certificates issued by certSIGN Enterprise CA Class 3 G2.

If a CRL entry is for a Root CA or Intermediate CA Certificate, including Cross Certificates, this ReasonCode CRL entry extension is always present, with an allowed value.

For the SSL/TLS certificates, the reasons of certificate revocation with code in **CRL Reason**:

1. No reason provided or unspecified (RFC 5280 CRLReason #0)
    - When the reason codes do not apply to the revocation request, the subscriber MUST NOT provide a reason code other than "unspecified".
2. keyCompromise (RFC 5280 CRLReason #1)
    - The certificate subscriber MUST choose the "keyCompromise" revocation reason when they have reason to believe that the private key of their certificate has been compromised, e.g. an unauthorized person has had access to the private key of their certificate.
3. affiliationChanged (RFC 5280 CRLReason #3)
    - The certificate subscriber SHOULD choose the "affiliationChanged" revocation reason when their Subject's name or other Subject Identity Information in the Certificate has changed, but there is no cause to suspect that the Certificate's Private Key has been compromised.
4. superseded (RFC 5280 CRLReason #4)
    - The certificate subscriber SHOULD choose the "superseded" revocation reason when the Certificate is being replaced because: the Subscriber has requested a new Certificate, the CA has reasonable evidence that the validation of domain authorization or control for any fully-qualified domain name or IP address in the Certificate should not be relied upon, or the CA has revoked the Certificate for compliance reasons such as the Certificate does not comply with the Baseline Requirements or the CA's CPS.
5. cessationOfOperation (RFC 5280 CRLReason #5)
    - The certificate subscriber SHOULD choose the "cessationOfOperation" revocation reason when the website with the Certificate is shut down prior to the expiration of the Certificate, or if the Subscriber no longer owns or controls the Domain Name in the Certificate prior to the expiration of the Certificate.
6. privilegeWithdrawn (RFC 5280 CRLReason #9)[3]

[3] *The privilegeWithdrawn reasonCode does not need to be made available to the certificate subscriber as a revocation reason option, because the use of this reasonCode is determined by the CA operator and not the subscriber.*

- The CRLReason privilegeWithdrawn is intended to be used when there has been a subscriber-side infraction that has not resulted in keyCompromise, such as the certificate subscriber provided misleading information in their certificate request or has not upheld their material obligations under the subscriber agreement or terms of use.

The Subscriber Agreement inform Subscribers about the revocation reason options listed above and provide explanation about when to choose each option. Revocation requests templates, that the CA provides to the Subscriber, allow for these options to be easily specified when the Subscriber requests revocation of their Certificate, with the default value being that no revocation reason is provided (i.e. the default corresponds to the CRLReason "unspecified (0)" which results in no reasonCode extension being provided in the CRL).

7.3 OCSP profile

The protocol of on-line certificate status verification (OCSP) allows certificate status evaluation.
OCSP service is provided by CERTSIGN on behalf of all affiliated Certification Authorities. OCSP server, which issues certificate status confirmations, employs a special key pair for each Intermediate CA and Root CA, generated exclusively for this purpose.
OCSP server certificate contains the extension extKeyUsage, described in RFC 5280.
This extension is set as non-critical, and means that a Certification Authority issuing the certificate to the OCSP server confirms with its signature delegation of the authorization to issue certificate status conformation (of this authority's Subscribers).
As well, OCSP server certificate contains the OCSPNoCheck extension, described by RFC 6960. This extension is declared non-critical which means that an OCSP client receives a response signed with the private key associated to this certificate can trust the OCSP server certificate status, without being necessary to check its revocation status.
The entity receiving a confirmation issued by the OCSP server must support the standard response format with **id-pkix-ocsp-basic** identifier.
Information about certificate status is included in **certStatus** field of **SingleResponse** structure. This may have one of the following three main values:
- GOOD – indicates the valid status of certificate
- REVOKED – indicates that the certificate was issued and revoked or the certificate was not issued in accordance with the RFC 6960
- UNKNOWN – indicates that there is not enough information to determine the certificate status

When the OCSP answer contains an error code (message), the answer is not digitally signed (RCF 6960).

7.3.1 Version numbers (s)

OCSP server operating within CERTSIGN issues certificate status confirmations in accordance with the RFC 6960. The only allowable value of the version number is 0 (it is an equivalent of v1 version).

7.3.2 OCSP extensions

In compliance with RFC 6960, CERTSIGN OCSP server accepts the following extension:
Nonce – binding a request and a response to prevent reply attacks. **Nonce** is included in **requestExtension** of the **OCSPRequest** and repeated in the field **responseExtension** of the **OCSPResponse**.

If an OCSP response is for a Root CA or Intermediate CA Certificate, including Cross Certificates, and that certificate has been revoked, then the **revocationReason** field within the **RevokedInfo** of the **CertStatus** is present, and contains a value permitted for CRLs, as specified in Section 7.2.2 above.
For end-user certificates requested to be revoked this extension is NOT used.