Public Key Service

Profil der nichtqualifizierten PKS-Zertifikate

Version: 2.1
Stand: 03.09.2019
Status: Freigegeben

Öffentliches Dokument

Business flexibility T · · Systems · · ·

Business flexibility T · · Systems · · ·

Impressum

Herausgeber
T-Systems International GmbH

Dateiname	Dokumentennummer	Dokumentenbezeichnung
PKS Zertifikatsprofil ECC und RSA2048 Ergänzung v2.1.docx		Spezifikation

Version	Stand	Status
2.1	03.09.2019	Freigegeben

Inhaltsverzeichnis

1 Einleitung

Die zusätzlich zu den qualifizierten Zertifikaten des Public Key Service ausgegeben nichtqualifizierten Zertifikate zur Authentisierung, Verschlüsselung und fortgeschrittenen elektronischen Signatur werden teilweise auch als Ergänzungszertifikate bezeichnet.

Sie sind konform zur COMMON-PKI Spezifikation [COMMON-PKI]. Damit ist eine Interoperabilität zu anderen Zertifizierungsstellen gegeben, die diesen Standard unterstützen, und die Zertifikate können von Standard-Clients verarbeitet werden.

Die COMMON-PKI-Spezifikation [COMMON-PKI] ist eine Profilierung der internationalen PKIX-Standards ist.

Dieses Dokument beschreibt den Aufbau der verschiedenen nichtqualifizierten PKS-Zertifikate. Die in dem jeweiligen Zertifikatstyp enthaltenen Daten werden ausgehend von der jeweils grundlegenden Struktur gemäß X.509 [X509] erläutert. Dabei werden die Felder und mögliche Inhalte, sowie die zu verwendenden Datentypen festgelegt.

2 Profil der nichtqualifizierten PKS-Zertifikate

Der generelle Aufbau der nichtqualifizierten PKS-Zertifikate entspricht der Norm X.509 und hat folgende Struktur:

```
Certificate ::= SEQUENCE
{
       toBeSigned     SEQUENCE {
         version      [0] Version DEFAULT v1
         serialNumber    INTEGER
         signature       AlgorithmIdentifier
         issuer       Name
         validity        Validity
         subject      Name
         subjectPublicKeyInfo    SubjectPublicKeyInfo
         issuerUniqueID    [1] IMPLICIT UniqueIdentifier
                                                  Optional
         subjectUniqueID   [2] IMPLICIT UniqueIdentifier
                                                  Optional
         extensions      [3] Extensions    Optional
         }
       signatureAlgorithm     AlgorithmIdentifier
       signatureValue      BIT STRING
}
```

Die Datentypen der verwendeten Felder des Zertifikates und deren mögliche Inhalte werden in den folgenden Kapiteln erläutert.

2.1 Übersicht über die Felder der nichtqualifizierten PKS-Zertifikate

Die folgende Tabelle zeigt alle möglichen Felder innerhalb des Zertifikates, die von PKS genutzt werden. Nicht verwendete Felder werden in der Tabelle nicht aufgeführt. Die Extensions werden zusätzlich mit P (Pflicht, immer vorhanden), B (bedingt Pflicht) oder O (optional) gekennzeichnet.

Name des Feldes	**Bedeutung**		**Wert**
Version	Versionsnummer für die verwendete Zertifikatsstruktur		2 (Version 3)
SerialNumber	Eindeutige Zertifikatsnummer innerhalb der CA		126 Bit langer Zufallswert
Signature	Algorithmus der Signatur des Zertifikates		ecdsa-with-SHA256 (1.2.840.10045.4.3.2)
Issuer	Name des Herausgebers des Zertifikates		z.B. C=DE / O=T-Systems International GmbH / CN=TeleSec PKS CA 8
Validity	Gültigkeitszeitraum des Zertifikates		z.B. vom 02.04.20019 09:20:23 bis zum 02.04.2021 23:59:00
Subject	Name des Inhabers des Zertifikates		z.B. C=DE / CN=Herbert Mustermann / SER=1
SubjectPublicKeyInfo	Öffentlicher Schlüssel des Zertifikatsinhabers mit dem zugehörigen Algorithmus		
SignatureAlgorithm	Algorithmus, mit dem das Zertifikat signiert worden ist		ecdsa-with-SHA256 (1.2.840.10045.4.3.2)
SignatureValue	Signatur des Zertifikates		BIT STRING
Extensions			
AuthorityKeyIdentifier	P	Informationen zur Identifizierung des öffentlichen Schlüssels des Ausstellerzertifikates	keyIdentifier = OCTET STRING mit einer Länge von 20 Bytes
SubjectKeyIdentifier	P	Informationen zur Identifizierung des öffentlichen Schlüssels des Zertifikates	OCTET STRING mit einer Länge von 20 Bytes

KeyUsage	P	Information, für welchen Zweck das Zertifikat benutzt werden darf	Verschlüsselungszertifikat: keyAgreement Authentisierungszertifikat: digitalSignature Signaturzertifikat: nonRepudiation
ExtendedKeyUsage	P	Information, für welchen Zweck das Zertifikat benutzt werden darf	Verschlüsselungszertifikat: Sichere E-Mail (1.3.6.1.5.5.7.3.4) Authentisierungszertifikat: Smartcard-Anmeldung (1.3.6.1.4.1.311.20.2.2) Clientauthentifizierung (1.3.6.1.5.5.7.3.2) Signaturzertifikat: Sichere E-Mail (1.3.6.1.5.5.7.3.4)
CertificatePolicy	P	Information, unter welchen Bedingungen das Zertifikat erstellt wurde und unter welchen Bedingungen es genutzt werden darf; Information, wo das Certification Practice Statement zu finden ist	Zertifizierungsrichtlinie für die T-Systems Trust Center Public Key Infrastrukturen 1.3.6.1.4.1.7879.13.18 Und die URL der CPS
SubjectAltName	P	Zusätzlicher technischer Name für den Inhaber des Zertifikates.	z.B. E-Mail-Adresse = Herbert.Mustermann@xyz.de
AuthorityInfoAccess	P	URLs für den Zugriff auf Statusinformationen per OCSP und zum Download des CA Zertifikats	OCSP: http://ocsp.pks.telesec.de/ocspr CA: http://crt.pks.telesec.de/crt/TeleSec_PKS_CA_8.crt

2.2 Das Feld Version

In dem Feld `Version` wird die Version der Datenstruktur des Zertifikates kodiert. Alle Zertifikate müssen die Versionsnummer v3 haben, da nur in dieser Version Extensions enthalten sein dürfen. Der Datentyp von `Version` ist INTEGER und hat immer den Wert 2 (entspricht v3).

2.3 Das Feld SerialNumber

In diesem Feld wird die Zertifikatsnummer abgelegt. Diese Zertifikatsnummer muss innerhalb des Zertifizierungsbereiches (gleicher Issuer) eindeutig sein. Der Datentyp der Zertifikatsnummer ist INTEGER und muss immer positiv sein. Die maximale Länge des kodierten Wertes darf 20 Bytes (1 <= Zertifikatsnummer < 2^{159}) nicht überschreiten.

2.4 Das Feld Signature

Das Feld `Signature` enthält den Bezeichner des Signaturalgorithmus, der von der CA für die Erstellung des Zertifikates benutzt wird. Der Inhalt dieses Feldes hat folgende Struktur:

```
AlgorithmIdentifier ::= SEQUENCE
{
    algorithm       OBJECT IDENTIFIER
    parameters      ANY DEFINED BY algorithm OPTIONAL
}
```

Folgender Algorithmus wird verwendet:

ecdsa-with-SHA256 (1.2.840.10045.4.3.2)

2.5 Das Feld Issuer

In dem Feld `Issuer` wird der Name des Herausgebers abgelegt. Der Inhalt dieses Feldes muss exakt mit dem Inhalt des Subject-Feldes des CA-Zertifikates übereinstimmen, von dem das Benutzerzertifikat unterschrieben worden ist.

Folgende Attribute werden verwendet:

- countryName (vorgeschrieben)
- organizationName (vorgeschrieben)
- organizationalUnitName (optional)
- commonName (vorgeschrieben)

Das Feld hat die folgende Datenstruktur:

```
Name ::= CHOICE {RDNSequence}

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE
{
 type       AttributeType
 value      AttributeValue
}

AttributeType ::== OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType
```

Anmerkung:

Der Inhalt des Feldes Issuer wird aus dem Subject-Feld des CA-Zertifikates entnommen von dem das Zertifikat unterschrieben ist.

2.6 Das Feld Validity

In diesem Feld wird der Gültigkeitszeitraum des Zertifikates eingetragen. Der Inhalt des Feldes hat folgende Syntax:

```
Validity ::= SEQUENCE
{
    notBefore     Time
    notAfter      Time
}

Time ::= CHOICE
{
    utcTime     UTCTime
    generalizedTime GeneralizedTime
}
```

Die Uhrzeit wird bis zum Jahr 2049 als UTCTime kodiert werden. Ab dem Jahr 2050 wird die Kodierung GeneralizedTime verwendet.

2.7 Das Feld Subject

Der Name des Zertifikatsinhabers wird in dem Feld `Subject` gespeichert. Der Name für den Zertifikatsinhaber muss innerhalb der Zertifizierungsstelle für die komplette Lebensdauer der CA eindeutig sein.

Folgende Attribute werden verwendet:

- countryName (vorgeschrieben)
- organizationName (optional)
- organizationalUnitName (optional)
- commonName (vorgeschrieben)
- serialNumber (vorgeschrieben)
- pseudonym (bedingt vorgeschrieben s.u.)
- surname (bedingt vorgeschrieben s.u.)
- givenname (bedingt vorgeschrieben s.u.)

Wenn der Zertifikatsinhaber ein Pseudonym als Name wünscht, wird zusätzlich das Attribut Pseudonym kodiert. Der Pseudonym-Name befindet sich immer in den Attributen commonName und pseudonym. Hierbei wird ein Pseudonym mit der Endung „:PN“ gekennzeichnet.

Die Attribute surname/givenname und pseudonym schließen sich gegenseitig aus. Wenn im Zertifikat ein Pseudonym eingetragen ist werden surname/givenname nicht verwendet. Wird kein Pseudonym verwendet so sind die Felder surname und givenname erforderlich.

Das Feld hat die gleiche Datenstruktur wie das Feld Issuer (siehe Kapitel 2.5).

2.8 Das Feld SubjectPublicKeyInfo

Der PublicKey des Zertifikatsinhabers wird in diesem Feld zusammen mit dem Algorithmus für den Gebrauch des Schlüssels gespeichert.

Der Inhalt des Feldes hat folgende Syntax:

```
SubjectPublicKeyInfo ::= SEQUENCE
{
    algorithm       AlgorithmIdentifier
    subjectPublicKey     BIT STRING
}
AlgorithmIdentifier ::= SEQUENCE
{
    algorithm       OBJECT IDENTIFIER
    parameters      ANY DEFINED BY algorithm OPTIONAL
}
```

Das Feld `parameters` kann zusätzliche Parameter des Algorithmus enthalten. Für Schlüssel basierend auf elliptischen Kurven wird hier die OID der Kurve eingetragen. Für diesen Zertifizierungsbereich wird die Kurve NIST-P-256 (OID {1 2 840 10045 3 1 7})verwendet. Die Codierung erfolgt als benannte Kurve.

2.9 Das Feld SignatureAlgorithm

Dieses Feld enthält den Algorithmus, mit dem das Zertifikat von der Zertifizierungsstelle unterschrieben worden ist. Der Inhalt und die Kodierung müssen identisch mit dem Feld `Signature` (siehe Kapitel 2.4) sein.

2.10 Das Feld SignatureValue

Das Feld `signatureValue` enthält die Signatur des Zertifikates, die von der Zertifizierungsstelle erzeugt worden ist.

Der Typ des Feldes `signatureValue` ist BIT STRING.

2.11 Das Feld Extensions

Die Extensions dienen zur Erweiterung der im Zertifikat enthaltenen Daten. Es gibt mehrere verschiedene Extensions, die in den folgenden Unterkapiteln aufgeführt werden.

Die Extensions haben folgende Syntax:

```
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
Extension ::= SEQUENCE
{
 extnId     OBJECT IDENTIFIER
 critical      BOOLEAN DEFAULT FALSE
 extnValue     OCTET STRING
}
```

Der Wert `extnId` gibt mit Hilfe eines Object Identifiers den Typ der in `extnValue` enthaltenen Extension an. Das Flag `critical` zeigt an, ob die Extension als kritisch markiert worden ist. Wenn `critical` auf TRUE gesetzt wird, bedeutet dies, dass der Client das Zertifikat als ungültig betrachten muss, wenn er die Extension nicht auswerten kann.

2.11.1 Die Extension AuthorityKeyIdentifier (vorgeschrieben)

Diese Extension dient zur eindeutigen Identifizierung des Ausstellerzertifikates, mit dem das Benutzerzertifikat unterschrieben worden ist.

Diese Extension wird als nicht kritisch markiert.

Die Extension AuthorityKeyIdentifier hat folgende Datenstruktur:

```
AuthorityKeyIdentifier ::= SEQUENCE
{
 keyIdentifier        [0] IMPLICIT KeyIdentifier OPTIONAL
 authorityCertIssuer     [1] IMPLICIT GeneralNames OPTIONAL
 authorityCertSerialNumber [2]    IMPLICIT    CertificateSerial-
 Number
                                                          OPTIO-
                                                             NAL
}
```

Das Feld `keyIdentifier` enthält einen eindeutigen Wert zur Identifizierung des öffentlichen Schlüssels des Herausgebenden CA-Zertifikats. Der Wert enthält den mit dem Algorithmus SHA-1 (160 bit) berechneten Hashwert über den subjectPublicKey des CA-Zertifikates (ohne die Bytes Tag, Länge und nicht benutzte Bits).

Die Felder `authorityCertIssuer` und `authorityCertSerialNumber` werden nicht verwendet.

2.11.2 Die Extension SubjectKeyIdentifier (vorgeschrieben)

Diese Extension dient zur eindeutigen Identifizierung des im Zertifikat enthaltenen öffentlichen Schlüssels.

Diese Extension wird als nicht kritisch markiert.

Die Extension SubjectKeyIdentifier hat folgende Datenstruktur:

```
SubjectKeyIdentifier ::= KeyIdentifier

KeyIdentifier ::= OCTET STRING
```

Der Wert der Extension ist der mit dem Algorithmus SHA-1 (160 bit) berechnete Hashwert über den subjectPublicKey (ohne die Bytes Tag, Länge und nicht benutzte Bits).

2.11.3 Die Extension KeyUsage (vorgeschrieben)

Mit dieser Extension wird festgelegt, für welchen Verwendungszweck der zum Zertifikat gehörende private Schlüssel benutzt werden darf.

Diese Extension wird als kritisch markiert.

Die Extension hat folgenden Aufbau:

```
KeyUsage ::= BIT STRING
{
 digitalSignature (0),
 nonRepudiation (1),
 keyEncipherment (2),
 dataEncipherment (3),
 keyAgreement (4),
 keyCertSign (5),
 crlSign (6),
 encipherOnly (7),
 decipherOnly (8)
}
```

Für

- Authentisierungszertifikate wird das Bit `digitalSignature`
- Verschlüsselungszertifikate für ECC Schlüssel enthalten das Bit `keyAgreement`.
- Signaturzertifikate wird nur das Bit `nonRepudiation`

gesetzt.

2.11.4 Die Extension ExtendedKeyUsage

Mit dieser Extension wird festgelegt für welchen Zweck das Zertifikat verwendet werden darf.

Diese Extension wird als nicht kritisch markiert.

Die Extension hat folgenden Aufbau:

```
extendedKeyUsage ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
```

Es werden die folgenden Werte eingetragen:

- Verschlüsselungszertifikate: Sichere E-Mail (1.3.6.1.5.5.7.3.4)
- Signaturzertifikate: Sichere E-Mail (1.3.6.1.5.5.7.3.4)
- Authentisierungszertifikate: Smartcard-Anmeldung (1.3.6.1.4.1.311.20.2.2), Clientauthentifizierung (1.3.6.1.5.5.7.3.2)

2.11.5 Die Extension CertificatePolicies (vorgeschrieben)

Diese Extension legt die Bedingungen fest, unter denen das Zertifikat herausgegeben wurde und unter denen es verwendet werden darf.

Diese Extension wird als nicht kritisch markiert.

```
CertificatePolicies ::= SEQUENCE SIZE (1..MAX) OF
                                                PolicyInformation
PolicyInformation ::= SEQUENCE
{
 policyIdentifier     CertPolicyId
policyQualifiers     SEQUENCE SIZE (1..MAX) OF
PolicyQualifierInfo OPTIONAL
}
CertPolicyId ::= OBJECT IDENTIFIER
PolicyQualifierInfo ::= SEQUENCE
{
 policyQualifierId PolicyQualifierId
 qualifier       ANY DEFINED BY policyQualifierId
}
PolicyQualifierId ::= OBJECT IDENTIFIER {id-qt-cps | id-qt-
unotice}
```

Als policyIdentifier wird immer “Zertifizierungsrichtlinie für die T-Systems Trust Center Public Key Infrastrukturen“, mit der OID 1.3.6.1.4.1.7879.13.18 verwendet. Die zusätzliche PolicyQualifierInfo enthält die URL wo die CPS zum Abruf bereitgehalten wird.

2.11.6 Die Extension SubjectAltNames (optional)

Mit Hilfe dieser Extension können alternative (technische) Namen für den Zertifikatsinhaber im Zertifikat eingefügt werden.

Diese Extension wird als nicht kritisch markiert.

Die Extension hat folgende Datenstruktur:

```
SubjectAltNames ::= GeneralNames
```

Wenn diese Extension vorhanden ist enthält sie genau eine E-Mail-Adresse des Zertifikatsinhabers als `rfc822Name`.

2.11.7 Die Extension AuthorityInfoAccess (vorgeschrieben)

Diese Extension enthält URLs für den Zugriff auf Statusinformationen per OCSP und zum Download des CA Zertifikats.

Diese Extension wird als nicht kritisch markiert.

Die Extension hat folgende Datenstruktur:

```
AuthorityInfoAccessSyntax  ::=

     SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription  ::=  SEQUENCE {

      accessMethod          OBJECT IDENTIFIER,

      accessLocation        GeneralName  }
```

Es werden folgende beiden Zugriffspunkte eingetragen:

- Onlinestatusprotokoll des Zertifikats (1.3.6.1.5.5.7.48.1):
  http://ocsp.pks.telesec.de/ocspr
- Zertifizierungsstellenaussteller (1.3.6.1.5.5.7.48.2):
  http://crt.pks.telesec.de/crt/TeleSec_PKS_CA_8.crt

Anhang A: Verwendete Attributtypen

Name des Attributs	Object Identifier	ASN.1 String Typ	maximale Länge
commonName	{id-at 3}	UTF8	64
surName	{id-at 4}	UTF8	64
givenName	{id-at 42}	UTF8	64
serialNumber	{id-at 5}	PrintableString	64
title	{id-at 12}	UTF8	64
organizationName	{id-at 10}	UTF8	64
organizationalUnit-Name	{id-at 11}	UTF8	64
BusinessCategory	{id-at 15}	UTF8	128
localityName	{id-at 7}	UTF8	128
stateOrProvinceNa-me	{id-at 8}	UTF8	128
countryName	{id-at 6}	PrintableString	2 (ISO 3166 Code)
distinguished-NameQualifier	{id-at 46}	PrintableString	64
initials	{id-at 43}	UTF8	64
generationQualifier	{id-at 44}	UTF8	64
eMailAddress	{pkcs-9 1}	IA5String	128
domainComponent	{0 9 2342 19200300 100 1 25}	IA5String	definiert in RFC 2247
postalAddress	{id-at 16}	SEQUENCE SIZE (1..6) OF UTF8	6 * 30, Verwendung wird in RFC 3039 beschrieben
pseudonym	{pkix 9 3}	UTF8	64
dateOfBirth	{id-pda 1}	GeneralizedTime	15
placeOfBirth	{id-pda 2}	UTF8	128
gender	{id-pda 3}	PrintableString SIZE (1)	Inhalt: „M“ oder „F“
countryOfCitizenship	{id-pda 4}	PrintableString	2 (ISO 3166 Code)
countryOfResidence	{id-pda 5}	PrintableString	2 (ISO 3166 Code)
nameAtBirth	{id-isismtt-at 14}	UTF8	64

Für die UTF8-Kodierung wird einen Auszug aus dem UTF8-Zeichensatz verwendet, der nur ANSI / ISO 8859-1 Zeichen (Unicode Latin-1 Seite) enthält. Andere Zeichen, die nicht in diesem Zeichensatz enthalten sind, dürfen nicht benutzt werden.

Besonderheiten für bestimmte Attribute:

- commonName: Ein Pseudonym wird immer mit der Endung „:PN“ als Common-Name eingefügt werden. Zusätzlich wird der gleiche Inhalt (einschließlich der Endung) in dem Attribut pseudonym eingefügt.

Abkürzungsverzeichnis

CA	Certification Authority
ECC	Elliptic Curve Cryptography
HTTP	Hypertext Transmission Protocol
LDAP	Lightweight Directory Access Protocol
OCSP	Online Certificate Status Protocol

Literaturverzeichnis

[COMMON-PKI] Common-PKI COMMON PKI SPECIFICATIONS FOR INTEROPERABLE APPLICATIONS VERSION 2.0 – 20 JANUARY 2009

[X509] Recommendation X.509: The Directory – Authentication Frame