

Einführung der Gesundheitskarte

Schnittstellen- und Prozessspezifikation – CVC- Root

G2 – CVC-Root

Version:	2.0.1
Revision:	-
Stand:	12.02.2019
Status:	Vorbereitet zur Freigabe
Klassifizierung:	vertraulich
Referenzierung:	[Atos_SchnittProz_CVC-ROOT]

Dokumentinformationen

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
1.0.0	04.12.2013	Alle	Zur Abstimmung freigegeben	Vennemann
1.1.0	15.01.2014	Alle	Überarbeitung gemäß gematik Kommentierung	Vennemann
1.2.0	18.02.2014	Alle	Überarbeitung gemäß gematik Kommentierung	Vennemann
1.3.0	13.03.2014	1.3 3.5	Überarbeitung gemäß gematik Kommentierung Anpassung auf [gemSpec_CVC_Root] Version 1.4.0	Vennemann
1.4.0	20.05.2014	3.5 Diverse	Detaillierung des Aufbaus der Request-Struktur Einarbeitung diverser Verdeutlichungen	Vennemann
1.5.0	10.10.2014	3.5, 3.6	Überarbeitung gemäß gematik Anmerkungen	Vennemann
1.6.0		3 4	Überarbeitung der Kapitel	Granitzki
2.0.0	29.01.2019		Dokumentenüberarbeitung im Rahmen des Atos- Projektüberganges von „Los 5“ nach „CVC-Root“	Granitzki
2.0.1	12.02.2019	2.1.5 2.2	Überarbeitung lt. Mängelliste gematik vom 05.02.2019 Anpassung Abbildung 1	Granitzki

Inhaltsverzeichnis

Dokumentinformationen	2
Dokumentenhistorie	2
Inhaltsverzeichnis	3
1 Einordnung des Dokuments	4
1.1 Zielsetzung	4
1.2 Zielgruppe	4
1.3 Abgrenzungen	4
2 Überblick	5
2.1 Akteure und Rollen	5
2.1.1 Anbieter der CVC-Root-CA.....	5
2.1.2 TSP-CVC.....	5
2.1.3 gematik.....	5
2.1.4 Kartenherausgeber	5
2.1.5 Personalisierer.....	6
2.2 Nachbarsysteme	6
3 Beantragung eines CVC-CA Zertifikats	7
3.1 Voraussetzungen	7
3.2 Beantragung eines CVC-CA Zertifikats	7
3.3 Schriftlicher Antrag	8
3.3.1 Berechnung des Fingerprint.....	9
3.4 Zertifikatserstellung	9
3.5 Aufbau des Zertifikatrequests	10
3.6 Formatierung von Request-Inhalten	13
4 Web-Portal	14
Anhang A	15
A1 – Abkürzungen	15
A2 – Tabellenverzeichnis	15
A3 – Abbildungsverzeichnis	16
A4 – Referenzierte Dokumente	16

1 Einordnung des Dokuments

1.1 Zielsetzung

Das vorliegende Dokument beschreibt die Schnittstellen und Prozesse zwischen der CVC-Root-CA und dem antragstellenden TSP-CVC. Es soll die Nutzer in die Lage versetzen, die Schnittstellen und Prozessabläufe für die Beantragung und Erstellung eines CVC-CA-Zertifikats durch die CVC-Root korrekt zu nutzen und zu bedienen. Die Ausstellung der CVC-Zertifikate durch die CVC-Root-CA erfolgt gemäß der Ausgabepolicy [Atos_Policy_CVC-ROOT].

1.2 Zielgruppe

Das Dokument richtet sich an Trust Service Provider CVC, die von der CVC-Root-CA Zertifikate beziehen wollen.

1.3 Abgrenzungen

In diesem Dokument werden ausschließlich die Prozesse und Schnittstellen für die Beantragung und Erstellung eines CVC-CA-Zertifikats der Generation 2 beschrieben. Die vorliegende Spezifikation ist gültig für die Produktivumgebung (PU) und für die Referenz- und Testumgebung (RU/TU), wenn nicht anders dargestellt.

Definitionen für die Durchführung der Eigenverantwortlichen Tests und Zulassungstests sind in dem Dokument [Atos_Test_Evt_Zul] beschrieben.

Nicht Bestandteil des vorliegenden Dokumentes sind die Festlegungen zu den Prozessen zur Zulassung der CVC-Root-CA sowie die Prozesse zur Zulassung und Registrierung eines TSP-CVC.

2 Überblick

Dieser Abschnitt beschreibt aus Sicht der CVC-Root die beteiligten Akteure und Rollen, die an der Zertifikatausstellung beteiligt sind, und gibt einen Überblick über die Nachbarsysteme der CVC-Root-CA.

2.1 Akteure und Rollen

2.1.1 Anbieter der CVC-Root-CA

Der Anbieter der CVC-Root-CA betreibt als technischer Dienstleister im Auftrage der gematik die CVC-Root-CA. Im Rahmen des Aufbaus der CVC-Root-CA wurde diese im Zuge eines organisatorischen Verfahrens von der gematik zugelassen. Die CVC-Root-CA generiert die CVC-CA-Zertifikate für die CVC-CAs der zweiten Ebene. Dabei stellt sie sicher, dass

- ein CVC-CA-Zertifikat nur für eine CVC-CA der zweiten Ebene generiert wird, falls der TSP-CVC aktuell gültig durch die gematik zugelassen sowie registriert ist und, sofern erforderlich, eine Qualifizierung für diese CVC-CA vorliegt und
- das Ausstellen eines CVC-CA-Zertifikats gemäß den Vorgaben aus Kapitel 3 erfolgt.

Der Anbieter der CVC-Root-CA veröffentlicht den aktuellen öffentlichen Schlüssel der CVC-Root-CA über ein Web-Portal. Auf Anfrage wird dieser Schlüssel auch schriftlich zugeschickt.

2.1.2 TSP-CVC

Ein TSP-CVC ist für das Generieren der CV-Zertifikate für eine Chipkarte (eGK, HBA, SMB, gSMC) zuständig. Ein TSP-CVC muss bei der gematik im Zuge eines organisatorischen Verfahrens zugelassen und die durch den TSP-CVC betriebenen CVC-CAs registriert werden. Ein TSP-CVC muss das CVC-CA Zertifikat zur Generierung der CV-Zertifikate für eine Chipkarte vom Anbieter der CVC-Root-CA beziehen.

2.1.3 gematik

Die gematik fungiert als Zulassungsinstanz und Registrierungsstelle für TSP-CVC sowie für den Anbieter der CVC-Root-CA und legt die Sicherheitsanforderungen fest. Die gematik stellt der CVC-Root-CA Informationen über zugelassene und registrierte TSP-CVC zur Verfügung, die berechtigt sind, CVC-CA-Zertifikate bei der CVC-Root-CA zu beziehen.

2.1.4 Kartenherausgeber

Kartenherausgeber (Leistungserbringerorganisationen (LEOs), Kostenträger (KTR) und Gerätehersteller) sind für die Herausgabe von eGK, HBA, SMC-B, gSMC-K und gSMC-KT zuständig. Diese beauftragen jeweils einen TSP-CVC zur Produktion der gewünschten CV-Zertifikate. Es dürfen nur solche TSP-CVCs beauftragt werden, für die aktuell eine gültige Zulassung der gematik vorliegt. Für die eingesetzten CVC-CAs verfügt der TSP-CVC darüber hinaus über eine bei der gematik vorgenommene Registrierung sowie ggf. über eine Qualifizierung seitens der kartenherausgebenden Organisation.

2.1.5 Personalisierer

Der Personalisierer ist bei der Produktion der Chipkarte für die sichere Einbringung der korrekten Schlüssel und Zertifikate in die Karte verantwortlich. Im Rahmen der Produktion einer Chipkarte (eGK, HBA, SMC-B, gSMC) bringen die Personalisierer u. a. den aktuellen öffentlichen Schlüssel der CVC-Root-CA in die Chipkarte ein, der von der CVC-Root-CA bereitgestellt wird.

2.2 Nachbarsysteme

Die Nachbarsysteme der CVC-Root-CA bestehen aus der gematik, den TSP-CVC sowie den Kartenherausgebern/ Personalisierern.

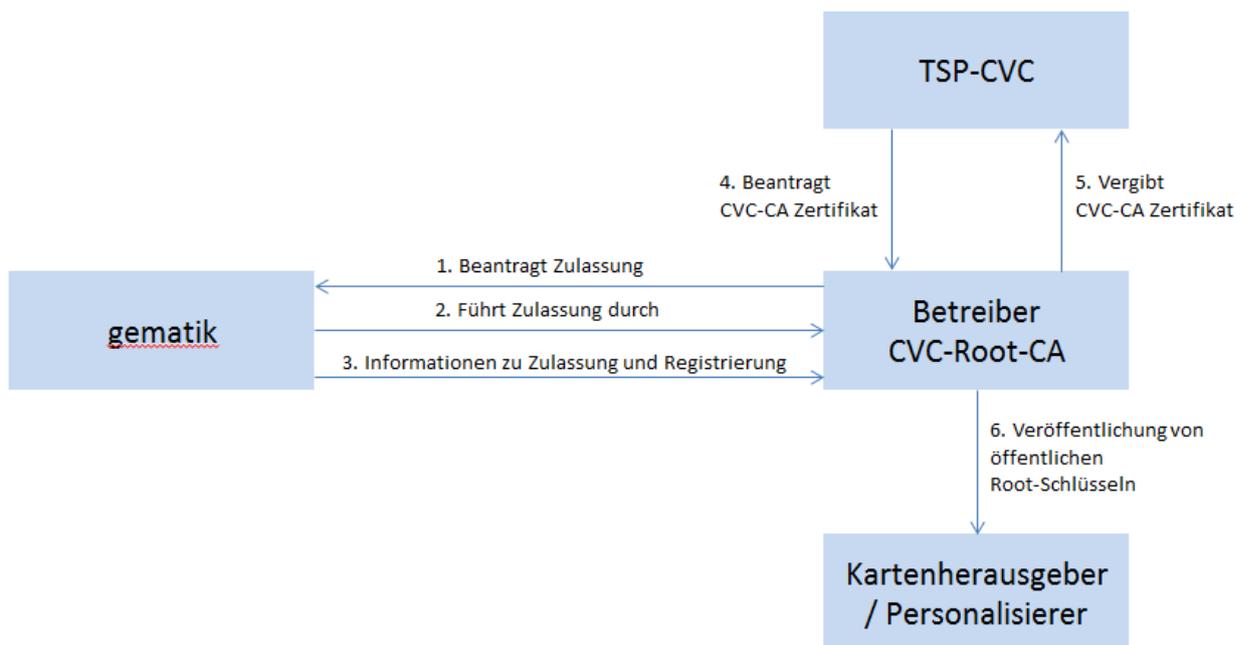


Abbildung 1 - Nachbarsysteme der CVC-Root-CA

Für die Prozesse der Zulassung und Registrierung (Schritte 1 bis 3) besteht eine organisatorische Schnittstelle zur gematik. Der Betreiber der CVC-Root-CA beantragt bei der gematik die Zulassung der CVC-Root (Schritt 1). Die gematik informiert den Betreiber über das Ergebnis des Zulassungsprozesses (Schritt 2). Die gematik informiert den Anbieter der CVC-Root-CA regelmäßig über die aktuell zugelassenen TSP-CVC und registrierten CVC-CAs (Schritt 3). Für die Erzeugung der CVC-CA-Zertifikate des TSP-CVC bestehen technische und organisatorische Schnittstellen (Schritte 4 und 5) zum Anbieter der CVC-Root-CA. Diese sind in Abschnitt 3 beschrieben.

Der öffentliche Schlüssel der CVC-Root-CA wird durch den Anbieter der CVC-Root-CA über ein Web-Portal veröffentlicht (Schritt 6). Kartenherausgeber bzw. Personalisierer benötigen den öffentlichen Root-Schlüssel für die Personalisierung der Karten.

3 Beantragung eines CVC-CA Zertifikats

Die Beantragung und Ausstellung von CVC-CA-Zertifikaten erfolgt gemäß [gemSpec_CVC_Root] und ist nachstehend näher erläutert.

3.1 Voraussetzungen

Der antragstellende TSP-CVC verfügt über eine Zulassung der gematik und hat die CVC-CA, für die er ein CVC-CA-Zertifikat beantragt, bei der gematik registriert. Die Informationen über die Zulassung und Registrierung werden der CVC-Root-CA durch die gematik zur Verfügung gestellt.

Nur für die Beantragung innerhalb der Referenz- und Testumgebung (RU/TU) kann die Zulassung entfallen.

3.2 Beantragung eines CVC-CA Zertifikats

Nachfolgend ist eine kurze Übersicht über die einzelnen Schritte des Beantragungsprozesses aufgeführt, die im weiteren Verlauf des Dokuments detailliert erläutert werden. Die Beantragung geschieht in den folgenden Schritten:

- Der TSP-CVC stellt einen schriftlichen Antrag bei der CVC-Root-CA.
- Nach erfolgreicher Prüfung des Antrags durch die CVC-Root-CA setzt diese sich mit dem TSP-CVC in Verbindung, um einen Termin zu vereinbaren, an dem ein Mitarbeiter des TSP-CVC das CVC-CA-Zertifikat persönlich bei der CVC-Root-CA abholen kann.
- An dem vereinbarten Termin überbringt ein Mitarbeiter des TSP-CVC den CVC-PKCS#10-Request persönlich zur CVC-Root-CA. Das Format des PKCS#10-Request ist in dem Abschnitt 3.5 beschrieben.
- Nach erfolgreicher Prüfung des CVC-PKCS#10-Requests wird durch die CVC-Root-CA das neue CVC-CA-Zertifikat erstellt (entsprechend den Vorgaben aus [gemSpec_PKI] und [gemSpec_Krypt]) und persönlich an den Mitarbeiter des TSP-CVC übergeben.

Das Vorgehen bei der Erzeugung von CVC-CA-Zertifikaten ist für die Test-CVC-Root-CA und für die Produktiv-CVC-Root-CA identisch. Mit dem Antrag muss der TSP-CVC jedoch angeben, ob ein Test-CVC-CA-Zertifikat oder Produktiv-CVC-CA-Zertifikat erzeugt werden soll. Die Abbildung 2 gibt eine schematische Übersicht über die Beantragung und Erstellung eines CV-Zertifikats.

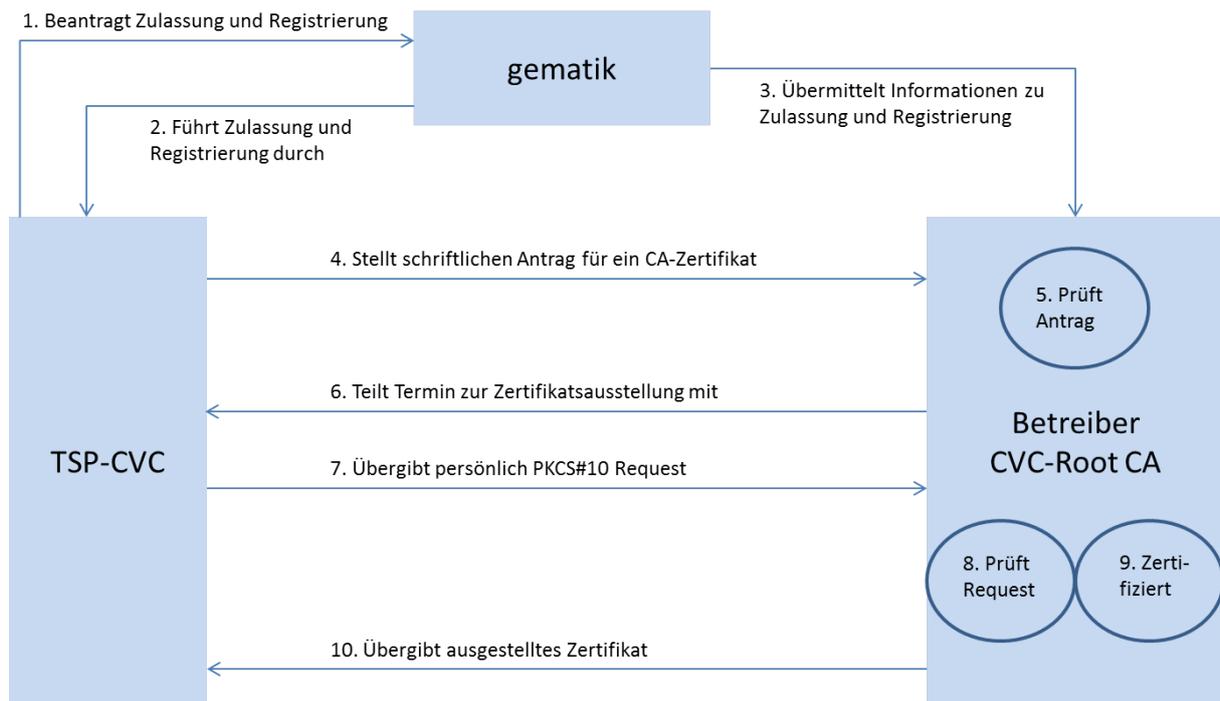


Abbildung 2 - Ablauf zur Beantragung eines CV-Zertifikats

3.3 Schriftlicher Antrag

Für den schriftlichen Antrag eines CVC-CA Zertifikats ist es erforderlich, das von der CVC-Root-CA bereitgestellte Formular auszufüllen und schriftlich an den Betreiber der CVC-Root-CA zu senden. Das Formular und die Adresse des Betreibers der CVC-Root-CA sind erhältlich unter:

<https://pki.atos.net/eqk>

Zur Beantragung eines CVC-CA Zertifikats müssen mindestens die nachfolgenden Informationen angegeben werden:

- Name und Anschrift der antragstellenden CVC-CA,
- Name und Vorname einer Kontaktperson, die im Rahmen der Zulassung (oder einer Änderung) der gematik benannt wurde und eine der Rollen "Leiter CVC-CA", "Sicherheitsbeauftragter" bzw. "Antragsteller CVC-CA-Zertifikat" inne hat,
- Typ des zu beantragenden CVC-CA-Zertifikats (Test oder produktiv),
- Fingerprint über den öffentlichen Schlüssel, für den das CVC-CA-Zertifikat erzeugt werden soll (siehe 3.3.1),
- Öffentlicher Schlüssel (Punkt Q) des zu zertifizierenden Schlüssels,
- Certificate Holder Referenz (CHR) zu dem zugehörigen Schlüssel, der zertifiziert werden soll und
- Unterschriften zweier hierfür berechtigter und bei der gematik registrierter Mitarbeiter des TSP-CVC. Eine der Unterschriften muss von einem Mitarbeiter stammen, dem die Rolle "Leiter CVC-CA" zugewiesen wurde und die zweite Unterschrift von einem

weiteren bei der Zulassung bzw. einer Änderungsmitteilung genannten Mitarbeiter ("Sicherheitsbeauftragter" bzw. "Antragsteller CVC-CA-Zertifikat").

Nach Eingang des Antrags bei dem Betreiber der CVC-Root-CA erfolgt die Prüfung der Antragsdaten und der Voraussetzungen des Antragsstellenden. Nach erfolgreicher Prüfung des Antrags durch Betreiber der CVC-Root-CA setzt dieser sich mit dem beantragenden TSP-CVC für die Terminabstimmung zur Zertifikatserstellung in Verbindung.

3.3.1 Berechnung des Fingerprint

Der Fingerprint über den öffentlichen Schlüssel wird durch die Berechnung des Hash-Wertes (SHA-256 nach [gemSpec_Krypt]) über den konvertierten Punkt Q (öffentlicher Schlüssel) in einen Oktett String gemäß „Uncompressed Encoding“ aus [BSI-TR-03111] generiert:

SHA-256(P2OS(Q))

Beispiel: Tabelle 1 enthält den öffentlichen Schlüssel P2OS(Q) eines 256 Bit ECDSA Schlüssels. Der darüber berechnete Fingerprint ist in Tabelle 2 enthalten.

```
04:5f:7e:cc:98:79:8f:a7:d2:0f:b9:c3:a6:02:43:
52:fc:70:be:7d:eb:22:58:b0:74:56:1a:73:ce:12:
75:2d:42:92:45:44:14:f0:cc:af:9d:66:17:57:fb:
d7:de:d5:c0:54:8e:41:b6:7c:6b:63:f9:fa:a9:c1:
46:4e:ee:7e:83
```

Tabelle 1 - Öffentlicher Schlüssel P2OS(Q)

Beispielhafte Berechnung des Fingerprints:

Annahme: Der in Tabelle 1 enthaltene Fingerprint befindet sich als Hexdump in der Datei `key.hex`. Der Fingerprint kann dann unter Linux mit der folgenden Befehlszeile berechnet werden:

```
cat key.hex | sed s/://g | xxd -r -p - | openssl dgst -c -sha256 -hex
```

```
fe:81:be:bf:38:bb:dd:e4:b1:be:60:a9:8b:a6:a5:
88:26:86:51:5f:7f:75:66:6c:78:78:4c:1e:05:31:
68:7e
```

Tabelle 2 – SHA-256 Fingerprint über den öffentlichen Schlüssel

3.4 Zertifikatserstellung

Die Erstellung des beantragten CVC-CA-Zertifikats erfolgt in den Räumlichkeiten des Root-CA-Betreibers. Hierfür ist eine persönliche Übergabe des CVC-PKCS#10-Requests erforderlich. Die Übergabe darf nur durch eine berechtigte Kontaktperson erfolgen, die sich als Mitarbeiter des TSP-CVC ausweisen kann.

Die Räumlichkeiten des Betreibers der CVC-Root-CA befinden sich bei der Atos Information Technology GmbH in Meppen.

3.5 Aufbau des Zertifikatrequests

Der TSP-CVC erstellt über den öffentlichen Schlüssel einen CVC-PKCS#10-Request gemäß der Struktur nach [gemSpec_CVC_Root] und [RFC2986]. Dieser wird durch den TSP-CVC mit dem zugehörigen privaten Schlüssel signiert.

Hinweis:

Die CVC-Root-CA verarbeitet nur base-64 kodierte CVC-PKCS#10-Request im PEM Format (umschlossen von „-----BEGIN CERTIFICATE REQUEST-----“ und „-----END CERTIFICATE REQUEST-----“) gemäß [RFC1421] und [RFC4648]. Hierbei hat ein Zeilenumbruch frühesten nach 64 Zeichen, spätestens nach 76 Zeichen zu erfolgen. Siehe hierzu auch das Beispiel in Tabelle 5.

Der PKCS#10-Request muss die folgenden Werte beinhalten:

- Im Request-Feld `certificationRequestInfo` muss `version` den Wert 0 haben.
- Im Request-Feld `certificationRequestInfo` muss `subject` die notwendigen Inhalte des CV-Zertifikats enthalten. Für die Angabe der Attribute müssen die OIDs gemäß Tabelle 3 verwendet werden. Die Kodierung der Werte innerhalb des Request muss hierbei als `PrintableString` nach [ITU-T X.680] erfolgen. Tabelle 4 beinhaltet hierzu Hinweise über die Formatierung der Eingangsdaten und die Länge der Datenfelder.

Hinweis: Aufgrund der Formatierung einiger Datenfelder gemäß Abschnitt 3.6 erfolgt eine differenzierte Darstellung der Länge im Gegensatz zu Tab_PKI_257 „Aufbau CAR“ aus [gemSpec_PKI].

- Im RequestInfo-Feld `subjectPKInfo` muss `algorithm` den Verwendungszweck `ecdsa-with-SHA256` (OID 1.2.840.10045.4.3.2), `ecdsa-with-SHA384` (OID 1.2.840.10045.4.3.3) oder `ecdsa-with-SHA512` (OID 1.2.840.10045.4.3.4) angeben.
- Im RequestInfo-Feld `subjectPKInfo` muss der Wert `AlgorithmIdentifer->parameters` den Typ des verwendeten öffentlichen Schlüssels enthalten (`brainpoolP256r1`, `brainpoolP384r1` oder `brainpoolP512r1`).
- Im RequestInfo-Feld `subjectPKInfo` muss `subjectPublicKey` den Domainparameter (Punkt Q des öffentlichen Schlüssels eines CV-Zertifikats der Kartengeneration 2; `brainpoolP256r1`, `brainpoolP384r1` oder `brainpoolP512r1`) enthalten.
- Im Request-Feld `signatureAlgorithm` muss `algorithm` das Signaturverfahren `ecdsa-with-SHA256` (OID 1.2.840.10045.4.3.2), `ecdsa-with-SHA384` (OID 1.2.840.10045.4.3.3) oder `ecdsa-with-SHA512` (OID 1.2.840.10045.4.3.4) angeben.

Die Angabe von Parametern (`subjectPKInfo`) bzw. die Verwendung des Signaturverfahrens (`signatureAlgorithm`) sind durch den TSP-CVC konsistent zu belegen bzw. anzuwenden. D. h. die Verwendung von `brainpoolP256r1` (`brainpoolP384r1` bzw. `brainpoolP512r1`) bedingt die Nutzung von `ecdsa-with-SHA256` (`ecdsa-with-SHA384` bzw. `ecdsa-with-SHA512`).

Die für das auszustellende Zertifikat notwendigen Attribute sind im `subject` des `CertificationRequestInfo` aufgeführt und müssen entsprechend der OID in der Tabelle 3 eingetragen sein (nach Vorgaben aus [gemSpec_CVC_Root]).

OID	Beschreibung
1.2.276.0.76.3.1.91.44.2.1	CA-Name
1.2.276.0.76.3.1.91.44.2.2	Service Indicator
1.2.276.0.76.3.1.91.44.2.3	Discretionary Data
1.2.276.0.76.3.1.91.44.2.4	Algorithm Reference
1.2.276.0.76.3.1.91.44.2.5	Aktivierungsjahr

Tabelle 3 - Inhalt Subject Zertifikatrequest CVC-Root

Feldname	Formatierung der Eingangsdaten	Länge des Datenfeldes innerhalb des Request
CA-Name	ASCII [RFC20]	5 Byte
Service Indicator	Nach Abschnitt 3.6	1 Byte
Discretionary Data	Nach Abschnitt 3.6	1 Byte
Algorithm Reference	Nach Abschnitt 3.6	2 Byte
Aktivierungsjahr	Nach Abschnitt 3.6	2 Byte

Tabelle 4 - Länge der Subject Datenfelder im Zertifikatrequest CVC-Root

Basierend auf den beschriebenen Inhalten enthält die Tabelle 5 ein Beispiel für einen PKCS#10-Request im PEM-Format (base-64 kodiert) und Tabelle 6 die ASN.1 Struktur des Request aus Tabelle 5.

```

-----BEGIN CERTIFICATE REQUEST-----
MIIBJjCBzgIBADBqMRYwFAYLKoIUAEwDAVssAgETBURFWVlZMRIwEAYLKoIUAEwD
AVssAgITATExEjAQBgsqghQATAMBWywCAxMBMDetMBEGCyqCFABMAwFbLAIIEwIw
MjETMBEGCyqCFABMAwFbLAIIFewIxNDBbMBUGCCqGSM49BAMCBgkrJAMDaggBAQcD
QgAEX37MmHmPp9IPucOmAkNS/HC+fesiWLB0VhpzzhJ1LUKSRUQU8MyvnWYXV/vX
3tXAVI5BtNxrY/n6qcFGTu5+g6AAMAoGCCqGSM49BAMCA0cAMEQCIBZvDI tGdVml
YOn8ls/1eIsc6js5bXGbqTnpcJ4vyVF+AiB+25b7JEEZx48gtiAiaLDny2yyC9L/
/acM/GPv/LDQxQ==
-----END CERTIFICATE REQUEST-----

```

Tabelle 5 - Beispiel PKCS#10-Request im PEM-Format

```

SEQUENCE {
  SEQUENCE {
    INTEGER 0
    SEQUENCE {
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER CA-Name (1 2 276 0 76 3 1 91 44 2 1)
          PrintableString 'DEYYY'
        }
      }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER Service Indicator (1 2 276 0 76 3 1 91 44 2 2)
          PrintableString '1'
        }
      }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER Discretionary Data (1 2 276 0 76 3 1 91 44 2 3)
          PrintableString '0'
        }
      }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER Algorithm Reference (1 2 276 0 76 3 1 91 44 2 4)
          PrintableString '02'
        }
      }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER Activation Year (1 2 276 0 76 3 1 91 44 2 5)
          PrintableString '14'
        }
      }
    }
  }
  SEQUENCE {
    SEQUENCE {
      OBJECT IDENTIFIER ecdsaWithSHA256 (1 2 840 10045 4 3 2)
      OBJECT IDENTIFIER brainpoolP256r1 (1 3 36 3 3 2 8 1 1 7)
    }
    BIT STRING
    04 5F 7E CC 98 79 8F A7 D2 0F B9 C3 A6 02 43 52
    FC 70 BE 7D EB 22 58 B0 74 56 1A 73 CE 12 75 2D
    42 92 45 44 14 F0 CC AF 9D 66 17 57 FB D7 DE D5
    C0 54 8E 41 B6 7C 6B 63 F9 FA A9 C1 46 4E EE 7E
    83
  }
  [0] {}
}
SEQUENCE {
  OBJECT IDENTIFIER ecdsaWithSHA256 (1 2 840 10045 4 3 2)
}
BIT STRING, encapsulates {
  SEQUENCE {
    INTEGER
    16 6F 0C 8B 46 75 59 A5 60 E9 FC 96 CF F5 78 8B
    1C EA 3B 39 6D 71 9B A9 39 E9 70 9E 2F C9 51 7E
    INTEGER
    7E DB 96 FB 24 41 19 C7 8F 20 B6 20 22 68 B0 E7
    CB 6C B2 0B D2 FF FD A7 0C FC 63 EF FC B0 D0 C5
  }
}
}

```

Tabelle 6 - Beispiel PKCS#10-Request mit SHA256

3.6 Formatierung von Request-Inhalten

Die Kodierung der Request-Inhalte erfolgt innerhalb des PKCS#10-Requests als PrintableString. Um binäre Inhalte von Daten als PrintableString darzustellen, muss eine Konvertierung dieser Inhalte entsprechend des Zeichenvorrats für PrintableString nach [ITU-T X.680] erfolgen. Hierzu muss jedes entsprechende Nibble der Eingangsdaten in den korrespondierenden ASCII-Wert nach Tabelle 7 umgewandelt werden.

Hexadezimaler Wertebereich	Korrespondierende ASCII Werte
0x0 – 0x9	0x30 – 0x39

Tabelle 7 - Konvertierung von Hexadezimalzahlen in ASCII-Werte

Beispiel:

- Die numerischen Inhalte eines CHR sind entsprechend nach Tabelle 7 wie folgt zu kodieren:

	Service Indicator	Discretionary Data	Algorithm Reference	Activation Year
BCD Format	1	0	02	14
Kodiert (nach Tabelle 7)	0x31	0x30	0x30 0x32	0x31 0x14

Diese Kodierung hat zur Folge, dass die Länge der konvertierten Ausgangsdaten die doppelte Länge der Eingangsdaten entspricht.

Seitens der CA werden die Eingangsdaten bei der Verarbeitung des Request zurückkonvertiert in die Ursprungsformate, um die korrekte Erstellung eines Zertifikats nach [gemSpec_PKI] sicherzustellen.

4 Web-Portal

Das von der CVC-Root-CA zur Verfügung gestellte Web-Portal ermöglicht dem antragstellenden TSP-CVC, unterschiedliche Informationen einzusehen und abzurufen, wie z. B:

- aktive und inaktive CVC-Zertifikate der Root-CA,
- die zugehörigen Fingerprints über die öffentlichen Schlüssel und
- das Antragsformular für die Erstellung eines Sub-CA Zertifikats durch die CVC-Root-CA.

Die Anmeldung an das Web-Portal erfolgt unter der folgenden URL:

<https://pki.atos.net/egk>

Anhang A

A1 – Abkürzungen

Kürzel	Erläuterung
ASCII	American Standard Code for Information Interchange
BCD	Binary Coded Decimals (binär kodierte Dezimalzahl)
CA	Certification Authority
CHR	Certificate Holder Referenz
CVC	Card Verifiable Certificate
eGK	Elektronische Gesundheitskarte
gSMC	Gerätebezogene Security Module Card
HBA	Heilberufsausweis
OID	Object Identifier, Objektkennung
PEM	Privacy Enhanced Mail
PKCS	Public Key Cryptography Standards
SMC-B	Security Module Card vom Typ B <medizinische Institution>
TSP	Trust Service Provider
URL	Uniform Ressource Locator

A2 – Tabellenverzeichnis

Tabelle 1 - Öffentlicher Schlüssel P2OS(Q).....	9
Tabelle 2 – SHA-256 Fingerprint über den öffentlichen Schlüssel	9
Tabelle 3 - Inhalt Subject Zertifikatrequest CVC-Root.....	11
Tabelle 4 - Länge der Subject Datenfelder im Zertifikatrequest CVC-Root	11
Tabelle 5 - Beispiel PKCS#10-Request im PEM-Format	11
Tabelle 6 - Beispiel PKCS#10-Request mit SHA256	12
Tabelle 7 - Konvertierung von Hexadezimalzahlen in ASCII-Werte	13

A3 – Abbildungsverzeichnis

Abbildung 1 - Nachbarsysteme der CVC-Root-CA 6
 Abbildung 2 - Ablauf zur Beantragung eines CV-Zertifikats 8

A4 – Referenzierte Dokumente

[Quelle]	Herausgeber: Titel
[Atos_Policy_CVC-ROOT]	Atos: Übergeordnete Ausgabe-Policy für CV-Zertifikate (ECC)
[Atos_Test_Evt_Zul]	Atos: Eigenverantwortliche Tests (EvT) und Zulassungstests in der RU/TU
[BSI-TR-03111]	BSI (2012): Elliptic Curve Cryptography, Version 2.0 https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03111/index.htm.html
[gemSpec_CVC_Root]	gematik: Spezifikation CVC - Root
[gemSpec_Krypt]	gematik: Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_PKI]	gematik: Übergreifende Spezifikation Spezifikation PKI
[ITU-T X.680]	International Telecommunication Union (ITU-T): X.680: Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation
[RFC20]	Network Working Group: ASCII format for Network Interchange
[RFC1421]	Network Working Group: Privacy Enhancement for Internet Electronic Mail – Part I: Message Encryption and Authentication Procedures; Abschnitt 4.3.2.4
[RFC2986]	Network Working Group: PKCS #10: Certification Request Syntax Specification; Version 1.7
[RFC4648]	Network Working Group: The Base16, Base32, and Base64 Data Encodings