*Beim vorliegenden Dokument handelt es sich um einen Entwurf der gematik in Vorbereitung auf zukünftige normative Festlegungen als Grundlage entsprechender Zulassungs- und Bestätigungsverfahren. Die gematik veröffentlicht diesen Entwurf mit dem Ziel, dass sich Interessierte bereits frühzeitig einen Überblick über die mögliche Weiterentwicklung der Telematikinfrastruktur verschaffen können. Die gematik übernimmt keine Gewähr für die Aktualität, Richtigkeit und Vollständigkeit dieses Entwurfes und behält sich das Recht vor, ohne vorherige Ankündigung Änderungen oder Ergänzungen vorzunehmen oder von den Regelungen insgesamt bzw. teilweise Abstand zu nehmen.*

Änderungsbedarf:

Ein Bedarf für die Behandlung von Attributzertifikaten besteht nicht mehr. Daher wird die Behandlung von Attributzertifikaten aus der Konnektorspezifikation und der HBA Kartenspezifikation entfernt. Da die Änderungen in den Schemadateien keine Wirkung auf Anfragen haben, die keine Attributzertifikate behandeln, und Attributzertifikate bisher nicht verwendet wurden, wird die Änderung als kompatibel bewertet und die Version der betroffenen Schemadateien in der dritten Stelle hochgezählt.

# Änderungen in gemSpec_Kon

4.1.8.3.1 TUC_KON_155 ”Dokumente zur Signatur vorbereiten”

**⌦ TIP1-A_4646 TUC_KON_155 „Dokumente zur Signatur vorbereiten“**

Der Konnektor MUSS den technischen Use Case TUC_KON_155 „Dokumente zur Signatur vorbereiten” umsetzen.

**Tabelle 1: TAB_KON_748 - TUC_KON_155 „Dokumente zur Signatur vorbereiten“**

| Element | Beschreibung |
|---|---|
| Name | TUC_KON_155 ”Dokumente zur Signatur vorbereiten” |
| Beschreibung | Die zu signierenden Dokumente werden entsprechend den Erfordernissen der Signaturverfahren für die QES oder nonQES vorbereitet. |
| Anwendungsumfeld | Erstellung von qualifizierten elektronischen Signaturen (QES) und nicht-qualifizierten elektronischen Signaturen (nonQES) |
| Auslöser | Aufruf durch TUC_KON_150 „Dokumente QES signieren“ oder TUC_KON_160 „Dokumente nonQES signieren“ |
| Vorbedingungen | keine |
| Eingangsdaten | • signatureMode<br>(Signaturart: QES \| nonQES)<br>• documentsToBeSigned<br>(Zu signierendes Dokument bzw. zu signierende Dokumente) und pro Dokument:<br>• documentFormat<br>(Formatangabe für das zu signierende Dokument)<br>• optionalInputs<br>(weitere optionale Eingabeparameter zur Steuerung der Details bei der |

| Element | Beschreibung |
|---|---|
| | zu erstellenden Signatur (siehe Operation SignDocument, Parameter dss:OptionalInputs), darin u.a.<br>▪ signatureType<br>(URI für den Signaturtyp XML-, CMS-, S/MIME-o PDF-Signatur)<br>• certificate<br>(Signaturzertifikat)<br>• ocspResponses – *optional*<br>(OCSP-Response des EE-Zertifikats, das bei der Signaturerstellung in die Signatur eingebettet wird.) |
| Komponenten | Konnektor |
| Ausgangsdaten | • preProcessedDocuments<br>(Aufbereitetes zu signierendes Dokument bzw. aufbereitete zu signierende Dokumente) |
| Standardablauf | <u>signatureType = XMLDSig (XAdES)</u><br>Entsprechend den Regeln für die QES und die nonQES werden zunächst weitere Signatureigenschaften zum jeweiligen Dokument in Form von `QualifyingProperties` (siehe [XAdES]) hinzugefügt. ~~Falls erforderlich, werden die Attributzertifikate hinzugefügt.~~<br><br>Die Systemzeit des Anwendungskonnektors muss in das XML-Element `SigningTime` (siehe [XAdES]) eingetragen werden. Die Signatur wird anschließend entsprechend [XMLDSig] vorbereitet. D. h., es wird je Dokument nach Erzeugung der Reference Elemente das SignedInfo Element aufgebaut. Dessen Inhalt ergibt dann nach erfolgter XML-Kanonisierung und Hashing die DTBS (Data To Be Signed), die später zur Karte gesendet werden.<br><br>certificate wird im Element `ds:KeyInfo/ds:X509Data` gespeichert.<br><br>Im Fall signatureMode = QES können neben den reinen Nutzdaten auch alle weiteren Elemente in die Signatur einbezogen werden, die für die Rekonstruktion der ursprünglich dargestellten Daten in der sicheren Anzeige erforderlich sind. Für XML-Dokumente sind das, falls vorhanden, das/die XML-Schema(ta). Für diese werden Referenzen (Hash + URI) in die Signatur eingebettet.<br><br>Die URI ist im Fall übergebener XML-Schemata der übergebene *signatureType* - Parameter. Die URI ist im Fall der im Konnektor im Rahmen einer Signaturrichtlinie hinterlegten XML-Schemata/XSL-Stylesheets die URI der Signaturrichtlinie, ergänzt um den Dateinamen mit Pfad, wie in der Signaturrichtlinie festgelegt.<br><br>(Beispiel: URI für Schemadatei CDA.xsd der Signaturrichtlinie SR_DF_SV_VHITG_ARZTBRIEF lautet:<br>urn:gematik:fa:sak:cda:r1:v1:fa:sak:cda:schemas:CDA.xsd)<br>Das Einbetten der Referenzen erfolgt über das XML-Element ds:object/ds:manifest (XMLDSig) mit eingebetteten XML-Elementen ds:Reference, die eine URI (RefURI) als Identifier für die jeweilige Datei und einen Hash über die jeweilige Resource enthalten.<br>Der ShortTextClientsystem muss in die Signatur in das `DataObjectFormat/Description`-Element gemäß [XAdES] (Abschnitt 7.2.5) eingebettet werden.<br><br>Falls durch den Aufrufparameter `SIG:IncludeRevocationInfo` angefordert, wird die für die Offline-Prüfung notwendige OCSP-Antwort im |

| Element | Beschreibung |
|---|---|
| | Sinne vom ES-X-L vom Konnektor in die Signatur eingebettet:<br>Die base-64 kodierte OCSP-Response wird im Feld `QualifyingProperties/UnsignedProperties/UnsignedSignatureProperties/RevocationValues/OCSPValues/EncapsulatedOCSPValue` (selbst DER-kodiert) gespeichert.<br><u>signatureType = CMS (CAdES)</u><br>Etwaig einzubettende XML-Schemata werden zunächst wie für XAdES definiert in ein ds:manifest-Element eingebettet. Die so erzeugte Zeichenkette wird als genau ein ASN.1 Character String vom Typ UTF8String verpackt. Dieser wird als contentDescription in einen Content-Hints Attributwert vom Typ ContentHints verpackt, wobei der contentType=id-data gemäß [CAdES].<br>Der ShortTextClientsystem muss in die Signatur in das `content-hints.ContentDescription`-Attribut gemäß [CAdES] (Abschnitt 5.10.3) eingebettet werden.<br>Ist die Einbettung von OCSP-Responses gefordert, wird die für die Offline-Prüfung notwendige OCSP-Antwort des EE-Zertifikats im Attribut SingedData.crls.other abgelegt.<br><u>signatureType = PDF/A (PAdES)</u><br>Der ShortTextClientsystem muss bei einer PDF-Signatur in das `Reason`-Feld eingebettet werden.<br>OCSP-Response des EE-Zertifikats ist im Document Security Store gemäß [PAdES-4] einzubetten.<br>Es sind die Vorgaben zum Signaturprofil gemäß Tabelle TAB_KON_779 „Profilierung der Signaturformate“ zu erfüllen.<br>Die aufbereiteten zu signierenden Dokumente werden an den Aufrufer zurückgegeben. |
| Varianten/Alternativen | keine |
| Fehlerfälle | Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_586 Fehlercodes TUC_KON_155 „Dokumente zur Signatur vorbe¬reiten“ „PDF/A (PAdES)“<br>Es ist nicht genügend Speicherplatz im PDF-Dokument verfügbar: 4205 |
| Nichtfunktionale Anforderungen | keine |
| Zugehörige Diagramme | keine |

**Tabelle 2: TAB_KON_586 Fehlercodes TUC_KON_155 „Dokumente zur Signatur vorbereiten“**

| Fehlercode | ErrorType | Severity | Fehlertext |
|---|---|---|---|
| 4205 | Technical | Error | Es ist nicht genügend Speicherplatz im PDF-Dokument verfügbar. |

⌫

4.1.8.3.1 TUC_KON_152 ”Signaturvoraussetzungen für QES prüfen”

**⌦ TIP1-A_4649 TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“**

Der Konnektor MUSS den technischen Use Case TUC_KON_152 „Signaturvoraussetzungen für QES prüfen” umsetzen.

**Tabelle 3: TAB_KON_751 – TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“**

| Element | Beschreibung |
|---|---|
| Name | TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“ |
| Beschreibung | Es werden die Voraussetzungen an die zu signierenden Dokumente und das Signaturzertifikat geprüft. Es werden die `QES_DocFormate` unterstützt. |
| Auslöser | TUC_KON_150 „Dokumente QES signieren“ |
| Vorbedingungen | keine |
| Eingangsdaten | • Zu signierende Dokumente<br>• optionale Eingabeparameter zur Steuerung der Details der Signaturerstellung<br>• cardSession Signaturkarte<br>• zu verwendende Identität (Zertifikatsreferenz)<br>• includeRevocationInfo [Boolean] - optional; Default: true (Dieser Parameter steuert die Einbettung von OCSP-Responses in die Signatur.<br>true: Die Sperrinformationen werden in ocspResponses zurückgegeben.) |
| Komponenten | Konnektor, Kartenterminal, Signaturkarte |
| Ausgangsdaten | • Prüfergebnis<br>• Signaturzertifikat<br>• ocspResponses - optional/nur wenn includeRevocationInfo = true (OCSP-Response des EE-Zertifikats, die beim Aufruf von TUC_KON_037 „Zertifikat prüfen“ zurückgegeben wird) |
| Standardablauf | 1. Abhängig von seinem Typ werden für jedes Dokument die typabhängigen Dokumentvalidierungsschritte durchgeführt (Aufruf TUC_KON_080 „Dokument validieren“). Wird der Aufruf von TUC_KON_080 mit einem Fehler beendet, wird die Prüfung im laufenden TUC mit diesem Fehler abgebrochen.<br>2. Durch Aufruf von TUC_KON_216 „Lese Zertifikat“ ~~werden~~ wird das Signaturzertifikat ~~und – falls für mindestens ein Dokument benötigt – die Attributzertifikate~~ von der Signaturkarte gelesen. |

| Element | Beschreibung |
|---|---|
| | 3. Das Signaturzertifikat ~~und die Attributzertifikate werden~~ wird durch Aufruf von TUC_KON_037 „Zertifikat prüfen“{<br>certificate = C.HP.QES;<br>qualifiedCheck = required;<br>offlineAllowNoCheck = true;<br>validationMode = OCSP;<br>getOCSPResponses = includeRevocationInfo~~;~~<br>~~attribCertificates = Liste der Attributzertifikate~~}<br>geprüft. |
| Varianten/Alternativen | keine |
| Fehlerfälle | keine |
| Nichtfunktionale Anforderungen | keine |
| Zugehörige Diagramme | keine |

**Tabelle 4: TAB_KON_588 Fehlercodes TUC_KON_152 „Signaturvoraussetzungen für QES prüfen“**

| Fehlercode | ErrorType | Severity | Fehlertext |
|---|---|---|---|
| Neben den Fehlercodes der aufgerufenen technischen Use Cases können keine weiteren Fehlercodes auftreten. | | | |

⌫

#### *4.1.8.4.6 TUC_KON_151 „QES Dokumentensignatur prüfen“*

⌦ **TIP1-A_4672 TUC_KON_151 „QES-Dokumentensignatur prüfen“**

Der Konnektor MUSS den technischen Use Case TUC_KON_151 „QES-Dokumentensignatur prüfen” umsetzen.

**Tabelle 215: TAB_KON_591 - TUC_KON_151 „QES-Dokumentensignatur prüfen“**

| Element | Beschreibung |
|---|---|
| Name | TUC_KON_151 „QES-Dokumentensignatur prüfen” |
| Beschreibung | Es wird die QES eines Dokuments geprüft. Dabei werden die Signaturverfahren laut Tabelle 187: TAB_KON_582 – Signaturverfahren unterstützt. Sind mehrere Signaturen vorhanden, so werden alle geprüft. Auch Parallel- und Gegensignaturen MÜSSEN unterstützt werden. |
| Eingangsanforderung | keine |
| Auslöser | Aufruf durch ein Clientsystem (Operation VerifyDocument) oder durch ein Fachmodul im Konnektor |
| Vorbedingungen | keine |
| Eingangsdaten | • signedDocument – *optional*<br>(QES-signiertes Dokument vom Typ `QES_DocFormate` |

| Element | Beschreibung |
|---|---|
| | -> siehe Definition in Operation `VerifyDocument` mit `SIG:Document`)<br>• signatureObject – *optional*<br>( -> siehe Definition in Operation VerifyDocument mit dss:SignatureObject.<br>Es werden Parallel- und Gegensignaturen unterstützt.)<br>• optionalInputParams<br>(optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs)<br>• certificates – *optional/falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden*<br>(X.509-Zertifikate ).<br>• xmlSchemas – *optional/nur für XML-Dokumente*<br>(XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata)<br>• includeRevocationInfo [Boolean]: – *optional; Default: false*<br>(Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur) |
| Komponenten | Konnektor |
| Ausgangsdaten | • verificationResult [VerificationResult]<br>(Ergebnis der Signaturprüfung)<br>• optionalOutput – *optional*<br>(weitere Ausgabedaten gemäß SIG:OptionalOutput) |
| Standardablauf | 1. **„DocumentValidation“**: Das signierte Dokument wird validiert mit Aufruf TUC_KON_080 „Dokument validieren“{ … }.<br>Treten Fehler bei der Validierung der Typkonformität auf, wenn die Signatur im Dokument eingebettet ist, wird die Prüfung mit einem Fehler abgebrochen.<br>Treten bei der Typkonformität, wenn die Signatur nicht im Dokument eingebettet ist, Fehler auf, so bricht der TUC nicht ab, sondern führt die folgenden Schritte soweit sinnvoll möglich durch. (Die Entscheidung über das sinnvoll Durchführbare liegt beim Hersteller des Konnektors.)<br>2. **„CoreValidation“**:<br>Es erfolgt die mathematische Prüfung der Signatur, bestehend aus der Prüfung der Hash-Kette bis zum signierten Hashwert und der Prüfung der Signatur unter Verwendung des öffentlichen Schlüssels, des Signaturwertes und des signierten Hashwertes.<br><u>XML-Signatur:</u><br>Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation.<br><u>CMS-Signatur:</u><br>Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652].<br><u>PDF-Signatur</u>:<br>Die Core Validation erfolgt entsprechend [PAdES-3] Kapitel 4.6 Signature Validation aus PAdES-BES Part 3.<br>Auch wenn die Validierung fehlschlägt, werden die folgenden Prüfschritte durchgeführt, so dass ein vollständiges Prüfprotokoll erstellt werden kann. |

| Element | Beschreibung |
|---|---|
| | 3. **„CheckSignatureCertificate“:**<br>**Teil 1: Signaturzertifikat ermitteln**<br>XML-Signatur:<br>Das Signaturzertifikat ist im XMLDSig Element `ds:KeyInfo/ds:X509Data` gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben.<br>CMS-Signatur:<br>Das Signaturzertifikat für CAdES ist im Feld `certificates` im `SignedData` Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben.<br>PDF-Signatur:<br>Das PDF Signaturzertifikat für PAdES ist im Feld SignedData.certificates entsprechend Kapitel 6.1.1 „Placements of the signing certificate“ [PAdES Baseline Profile] gespeichert oder wird als Eingangsparameter übergeben.<br>**Teil 2: Signaturzeitpunkt bestimmen**<br>Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_Eingebettet` wird wie folgt selektiert:<br>XML-Signatur:<br>Das XML element `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 7.2.1 XAdES [XAdES].<br>CMS-Signatur:<br>Das Attribut `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS].<br>PDF-Signatur:<br>Der Signaturzeitpunkt kann dem M Eintrag des Signature Dictionary entnommen werden [PAdES Baseline Profile] Kapitel 6.2.1 Signing time.<br>Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_Qualifiziert` wird wie folgt selektiert:<br>XML-Signatur:<br>Das XML element `SignatureTimeStamp` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 4.4.3.1 XAdES [XAdES].<br>CMS-Signatur und PDF-Signatur:<br>Das Attribut `signature-time-stamp` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 6.1 CAdES [CAdES].<br>Der Signaturzeitpunkt `Benutzerdefinierter_Zeitpunkt` liegt gegebenenfalls als Aufrufparameter vor. Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_System` wird ermittelt.<br>**Teil 3: Signaturzertifikatsprüfung:**<br>Bei der folgenden Signaturzertifikatsprüfung sind die Signaturzeitpunkte gemäß [TIP1-A_5540] zu berücksichtigen.<br>Die Signaturzertifikatsprüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen“ {<br>certificate = C.HP.QES; |

| Element | Beschreibung |
|---|---|
| | qualifiedCheck = required;<br>baseTime = Signaturzeitpunkt;<br>offlineAllowNoCheck = true;<br>validationMode = OCSP;<br>ocspResponses = OCSP-Response;<br>getOCSPResponses = includeRevocationInfo~~;~~<br>~~attribCertificates = Liste der Attributzertifikate~~<br>}.<br>Sind OCSP-Responses in der Signatur eingebettet, ist die jüngsten OCSP-Response des EE-Zertifikats, die für die Zertifikatsprüfung notwendig ist, beim Aufruf von TUC_KON_037 zu übergeben.<br>Sofern der Aufruf von TUC_KON_037 ocspResponses zurückgibt, wird die OCSP-Response des EE-Zertifikats in die Signatur eingebettet.<br>~~Sofern Attributzertifikate in der Signatur vorhanden sind, werden diese beim Aufruf von TUC_KON_037 übergeben.~~<br>Auch wenn die Zertifikatsprüfung fehlschlägt, werden die folgenden Prüfungen durchgeführt.<br>4. **„CheckPolicyConstraints“**:<br>In diesem Schritt wird das signierte Dokument entsprechend der Profilierung der Signaturformate (siehe Anhang B.2) geprüft. Es sind die Vorgaben für die Prüfung von Signaturen aus den Standards für AdES [XAdES], [XAdES Baseline], [CAdES], [CAdES Baseline], [PAdES-3] und [PAdES Baseline] umzusetzen. Dabei sind die Vorgaben aus Tabelle TAB_KON_779 „Profilierung der Signaturformate“ zu erfüllen.<br>Auch wenn nicht alle Anforderungen an das Format des signierten Dokuments erfüllt werden, wird die Prüfung mit den folgenden Schritten fortgesetzt, um ein vollständiges Prüfungsprotokoll zu erhalten.<br>5. Das Prüfergebnis (VerificationResult, OptionalOutput) wird an den Aufrufer zurückgegeben (siehe TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur). |
| Varianten/Alternativen | Keine |
| Fehlerfälle | Das Verhalten des TUCs bei einem Fehlerfall ist in TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen“ beschrieben.<br>(→ 2 **„CoreValidation“**) Interner Fehler: 4001, Signatur des Dokuments ungültig: 4115<br>(→3 **„CheckSignatureCertificate“**) Interner Fehler: 4001, Signaturzertifikat ermitteln ist fehlgeschlagen: 4206.<br>(→4 **„CheckPolicyConstraints“**) Interner Fehler: 4001, Dokument nicht konform zu Regeln für QES: 4124, Dokument nicht konform zu Profilierung der Signaturformate: 4208. |
| Nichtfunktionale Anforderungen | Keine |
| Zugehörige Diagramme | keine |

**Tabelle 216: TAB_KON_592 Fehlercodes TUC_KON_151 „QES Dokumentensignatur prüfen“**

| Fehlercode | ErrorType | Severity | Fehlertext |
|---|---|---|---|
| Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten: | | | |
| 4001 | Technical | Error | interner Fehler |
| 4115 | Security | Error | Signatur des Dokuments ungültig. Prüfung der Hashwertkette bzw. Prüfung der kryptographischen Signatur fehlgeschlagen. |
| 4124 | Technical | Error | Dokument nicht konform zu Regeln für QES |
| 4206 | Technical | Error | Signaturzertifikat ermitteln ist fehlgeschlagen |
| 4208 | Technical | Error | Dokument nicht konform zu Profilierung der Signaturformate |

Das Gesamtergebnis (VerificationResult) für die Prüfung einer Dokumentensignatur fasst die Ergebnisse der obigen Einzelschritte zusammen. Welche nicht erfüllten Kriterien einzelner Prüfschritte zu einem Fehler oder einer Warnung führen, wird durch die Regeln für die QES (siehe Anhang B.2) bestimmt.

**Tabelle 217: TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur**

| VerificationResult pro Signatur | | |
|---|---|---|
| | **Wert** | **Bedeutung** |
| | VALID | Die Signatur wurde gemäß den Regeln für die QES geprüft und für gültig befunden. |
| | INVALID | Die Signatur des Dokuments ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden. |
| | INCONCLUSIVE | Die Signatur wurde gemäß den Regeln für die QES geprüft. Allerdings konnten einige Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail.<br>Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich. |

| VerificationResult für gesamtes Dokument | |
|---|---|
| **Wert** | **Bedeutung** |
| VALID | Wenn VerificationResult für alle Signaturen zum Dokument VALID |
| INVALID | Wenn VerificationResult für eine Signatur zum Dokument INVALID |
| INCONCLUSIVE | in allen anderen Fällen |

Das VerificationResult pro Signatur MUSS wie folgt zur Belegung des Elements VerficationReport/IndividualReport/Result korrespondieren:

- wenn ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success AND ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
- INVALID, wenn ResultMajor UNGLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success ODER

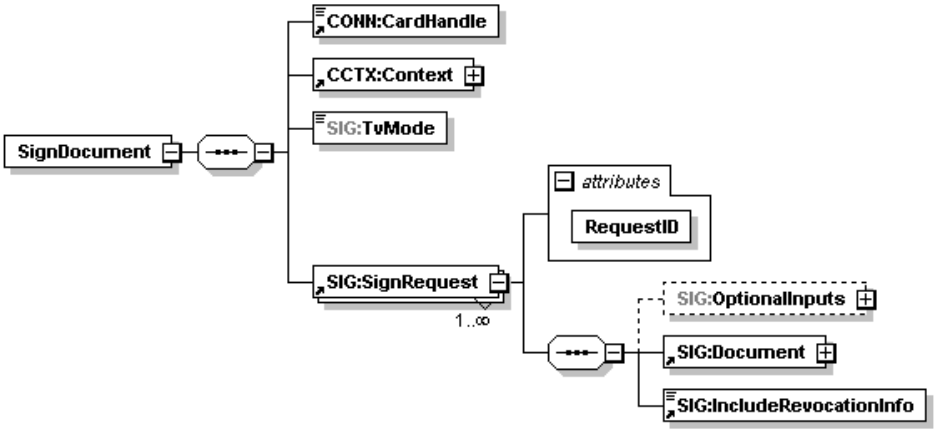
(ResultMajor GLEICH urn:oasis:names:tc:dss:1.0:resultmajor:Success UND ResultMinor GLEICH urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature)⌫

#### 4.1.8.5.1 SignDocument (nonQES und QES)

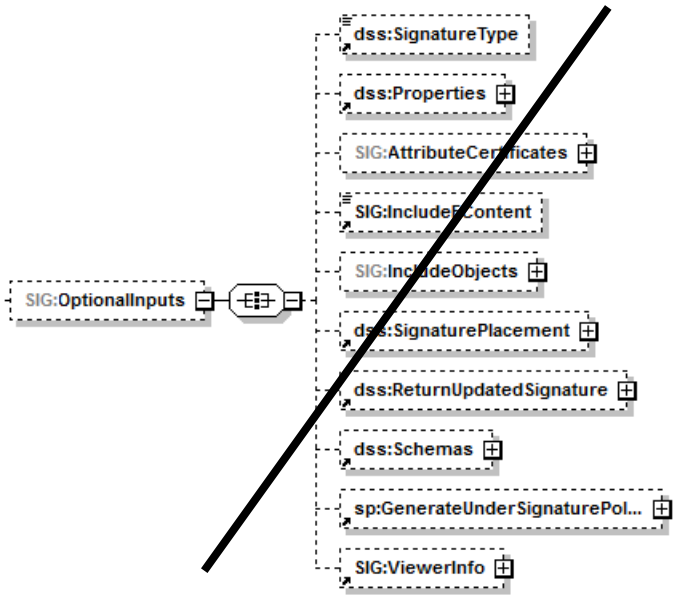
**⌦ TIP1-A_5010 Operation SignDocument (nonQES und QES)**

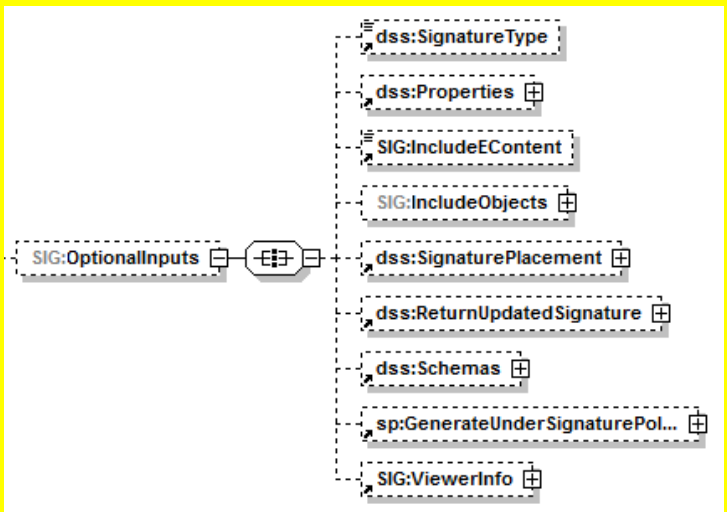
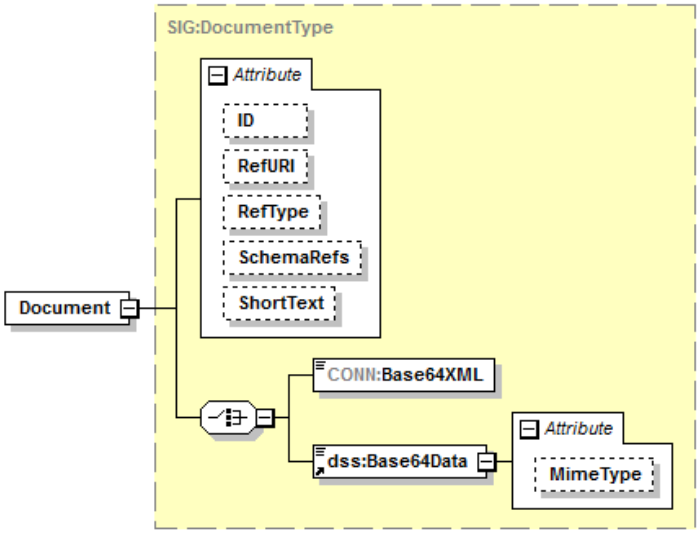
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation SignDocument anbieten**.**

**Tabelle 5: TAB_KON_065 Operation SignDocument (nonQES und QES)**

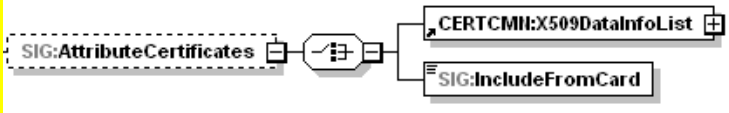
| | |
|---|---|
| **Name** | `SignDocument` |
| **Beschreibung** | Diese Operation lehnt sich an [OASIS-DSS] an. Sie enthält voneinander unabhängige SignRequests. Jeder SignRequest erzeugt eine Signatur für ein Dokument.<br>Für die qualifizierte elektronische Signatur (QES) werden die `QES_DocFormate` unterstützt. Für nicht-qualifizierte elektronische Signaturen (nonQES) werden die `nonQES_DocFormate` unterstützt.<br>Zur Signaturerzeugung werden Schlüssel und Zertifikate einer Chipkarte benutzt.<br>Unterstützte Karten sind für die QES der HBAx mit dem QES-Zertifikat. Für die nonQES wird für die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur“ die SM-B mit dem OSIG–Zertifikat unterstützt.<br>Bei der Erstellung von XML-Signaturen MUSS Canonical XML 1.1 verwendet werden [CanonXML1.1].<br>Es soll der Common-PKI-Standard eingesetzt werden, siehe [Common-PKI].<br>In Summe für die Größe der Dokumente in allen SignRequests innerhalb einer SignDocument-Anfrage MUSS der Konnektor eine Gesamtgröße von <= 250 MB unterstützen. |
| **Aufruf-parameter** |  |

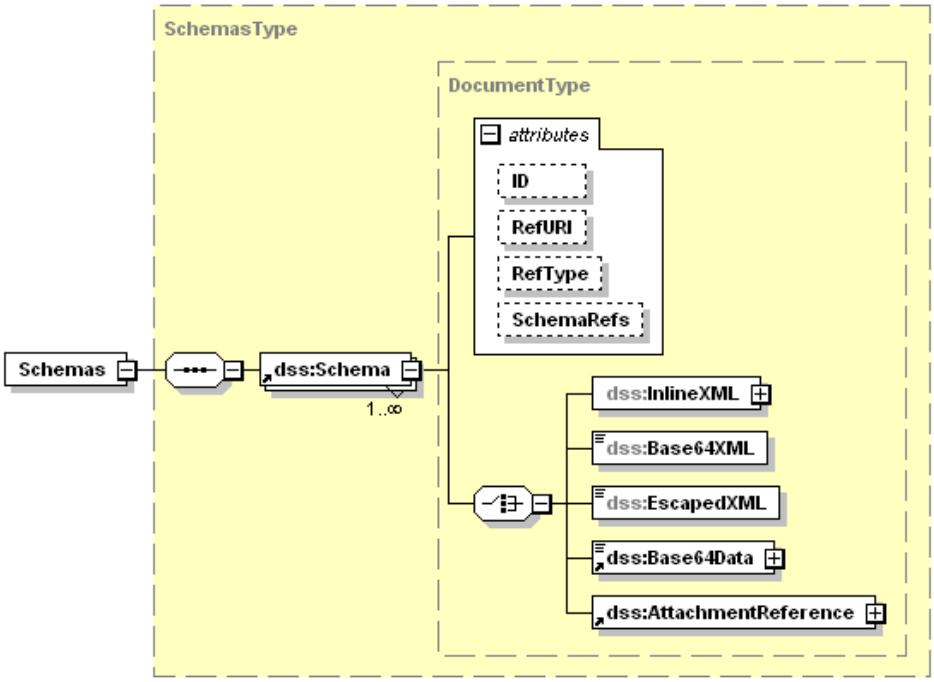
| Name | Beschreibung |
|---|---|
| `CONN:CardHandle` | Identifiziert die zu verwendende Signaturkarte.<br>Die Operation DARF die Signatur mit der eGK NICHT unterstützen. Wird die Operation mit einem nicht unterstützten Kartentypen aufgerufen, so MUSS der Konnektor die Bearbeitung mit dem Fehler 4126 abbrechen. |

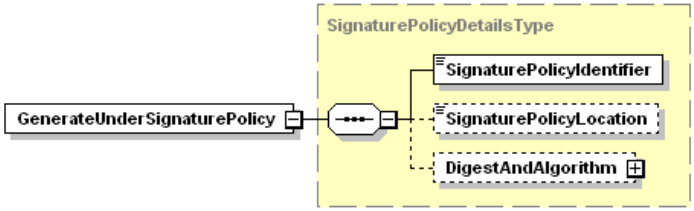
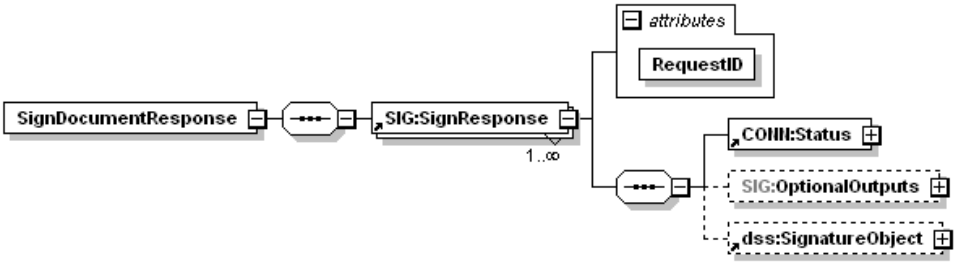
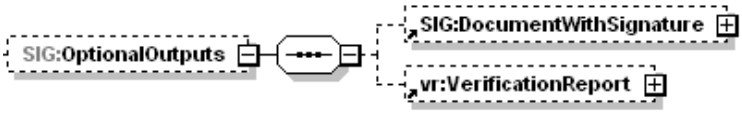
| | | |
|---|---|---|
| | `CCTX:Context` | <u>Aufrufkontext QES mit HBAx</u>:<br>MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend<br><u>Aufrufkontext nonQES mit SM-B</u>:<br>MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet |
| | `TvMode` | Der Parameter wird im Konnektor nicht ausgewertet. |
| | `SIG: JobNumber` | Die Nummer des Jobs, unter der der nächste Signaturvorgang gestartet wird.<br>Parameter muss gefüllt werden. |
| | `SIG:SignRequest` | Ein `SignRequest` kapselt den Signaturauftrag für ein Dokument.<br>Das verpflichtende XML-Attribut `RequestID` identifiziert einen `SignRequest` innerhalb eines Stapels von `SignRequests` eindeutig. Es dient der Zuordnung der `SignResponse` zum jeweiligen `SignRequest`. |
| | `SIG:OptionalInputs` | Enthält optionale Eingangsparameter (angelehnt an `dss:OptionalInputs` gemäß [OASIS-DSS] Section 2.7):<br>  |

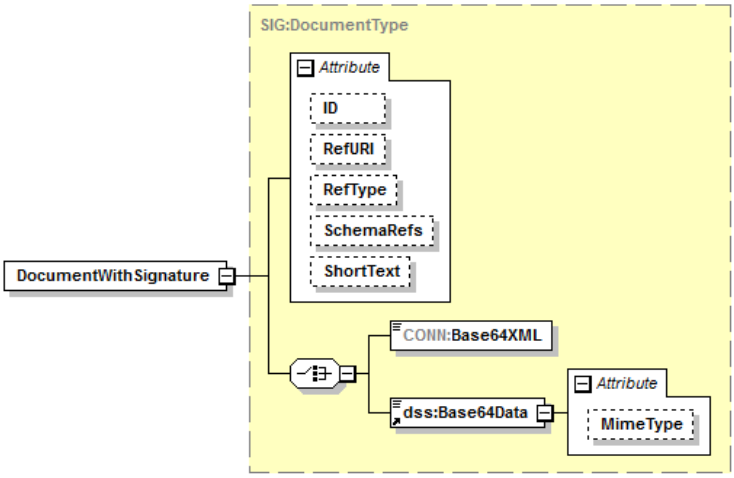
| | | |
|---|---|---|
| | |  |
| | `SIG:Document` |  <br>Dieses an das `dss:Document` Element aus [OASIS-DSS] Section 2.4.2 angelehnte Element enthält das zu signierende Dokument, wobei die Kindelemente `CONN:Base64XML` und `dss:Base64Data` auftreten können.<br>Bei den als `dss:Base64Data` übergebenen Dokumenten werden folgende (Klassen von) MIME-Types unterschieden:<br>”application/pdf-a” – für PDF/A-Dokumente,<br>”text/plain”,<br>”text/plain; charset=iso-8859-15” oder<br>”text/plain; charset=utf-8” – für Text-Dokumente,<br>”image/tiff” – für TIFF-Dokumente und<br>ein beliebiger anderer MIME-Type für nicht näher unterschiedene Binärdaten des spezifizierten Typs.<br>Der MIME-Type „text/plain“ wird interpretiert als „text/plain; charset=iso-8859-15”.<br>Das Element enthält ein Attribut `ShortText`. Es muss für QES-Signaturen bei jedem Aufruf vom Clientsystem übergeben werden, für nonQES-Signaturen ist es optional.<br>Über das Attribut `RefURI` kann gemäß [OASIS-DSS] (Abschnitt 2.4.1) ein zu signierender Teilbaum eines XML- |

| | | |
|---|---|---|
| | | Dokuments ausgewählt werden. |
| | `SIG:IncludeRevocationInfo` | Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen anfordern. Es wird ausschließlich die zu erstellende Signatur betrachtet, d.h. es erfolgt keine Einbettung von Sperrinformationen für bereits enthaltene Signaturen.<br>Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nicht unterstützt. |
| | `dss:SignatureType` | Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen spezifiziert werden. Hierbei MÜSSEN folgende Signaturtypen unterstützt werden:<br>**XML-Signatur**<br>Durch Übergabe der URI urn:ietf:rfc:3275 wird die Erstellung von XML-Signaturen gemäß [RFC3275], [XMLDSig] angestoßen.<br>Das zu verwendende Profil ist XAdES-BES ([XAdES]).<br>Die Rückgabe einer solchen Signatur erfolgt als `ds:Signature`-Element.<br>**CMS-Signatur**<br>Durch Übergabe der URI urn:ietf:rfc:5652 wird eine CMS-Signatur gemäß [RFC5652] angestoßen. Das zu verwendende Profil ist CAdES-BES ([CAdES]).<br>Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert.<br>**S/MIME-Signatur**<br>Durch Übergabe der URI „urn:ietf:rfc:5751" wird eine S/MIME-Signatur gemäß [RFC5751] angestoßen.<br><br>Die CMS-Signatur der übergebenen MIME-Nachricht erfolgt konform der Vorgaben zur CMS-Signatur. Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime" mit einer CMS-Struktur vom Typ SignedData.<br>Ist das übergebene Dokument keine MIME-Nachricht, so wie der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.<br>**PDF-Signatur**<br>Durch Übergabe der URI http://uri.etsi.org/02778/3 wird die Erzeugung einer PAdES-Basic Signatur gemäß [PAdES-3]angestoßen, wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird.<br>Handelt es sich beim übergebenen Dokument nicht um ein `Base64Data`-Element mit MIME-Type „application/pdf-a", so wird ein Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert.<br>Andere `SignatureType`-Angaben führen zu einer Fehlermeldung 4111 (Ungültiger Signaturtyp oder Signaturvariante).<br>Die Signaturtypen „XML-Signatur, CMS-Signatur, PDF-Signatur, S/MIME-Signatur" DÜRFEN für QES der HBAx nur mit dem QES-Zertifikat erfolgen, für nonQES nur mit dem OSIG-Zertifikat der SM-B. In jedem diese Anforderung verletzenden Fall MUSS der Fehler 4058 (Aufruf nicht |

| | | |
|---|---|---|
| | | zulässig) zurückgeliefert werden.<br>Fehlt dieses Element, so wird der Signaturtyp gemäß TAB_KON_583 – Default-Signaturverfahren aus dem Dokumententyp abgeleitet. |
| | `dss:Properties` | Durch dieses in [OASIS-DSS] (Abschnitt 3.5.5) definierte Element können zusätzliche signierte und unsignierte Eigenschaften (Properties) bzw. Attribute in die Signatur eingefügt werden.<br>Unterstützt werden genau folgende Attribute:<br>Im CMS-Fall (SignatureType = urn:ietf:rfc:5652) kann es XML-Elemente<br>`./SignedProperties/Property/Value/CMSAttribute`<br>und<br>`./UnsignedProperties/Property/Value/CMSAttribute`<br>enthalten. Ein solches XML-Element `CMSAttribute` muss ein vollständiges, base64/DER-kodiertes ASN.1-`Attribute` enthalten, definiert in [CMS#5.3.SignerInfo Type]. Es muss bei der Erstellung des CMS-Containers unverändert unter `SignedAttributes bzw. UnsignedAttributes` aufgenommen werden. |
| | ~~`SIG:AttributeCertificates`~~ |  <br>~~Dieses Element ist nur im Fall einer QES-Signatur relevant.~~<br>~~Ist dieses Element nicht vorhanden, werden keine Attributzertifikate eingebetttet.~~<br>~~Ist dieses Element vorhanden, können über das Element CERTCMN:X509DataInfoList Attributzertifikate beim Aufruf in dem Format mitgegeben werden, wie es die Operation ReadCardCertificate des Zertifikatsdienstes liefert.~~<br>~~Alternativ kann der Konnektor über das Element SIG:IncludeFromCard angewiesen werden, alle Attributzertifikate zum Basiszertifikat von der signierenden Karte zu lesen.~~<br>~~Die Zertifikate bettet der Konnektor für XAdES unter~~<br>~~`ds:Signature/ds:Object/QualifyingProperties/SignedProperties/SignedSignatureProperties/SignerRole/CertifiedRoles/CertifiedRole`~~<br>~~in die Signatur ein und für CAdES und PAdES unter~~<br>~~`SignedData/signerInfos/SignerInfo/SignedAttrs/SignerAttributes/CertifiedAttributes`~~<br>~~Es muss mindestens eine Anzahl von 10 Attributzertifikaten unterstützt werden.~~ |
| | `SIG:IncludeEContent` | Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden.<br>Die Verwendung dieses Parameters bei anderen Signaturtypen führt zu einem Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante). |
| | `SIG:IncludeObject` | Dieses Element enthält zum Anfordern einer Enveloping XML Signatur ein dss:IncludeObject-Element gemäß [OASIS-DSS] (Abschnitt 3.5.6).<br>Ist das Element vorhanden und ein anderer Signaturtyp als |

| | | |
|---|---|---|
| | | eine XML-Signatur angefordert, so wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. |
| | `dss:SignaturePlacement` | Durch dieses in [OASIS-DSS] (Abschnitt 3.5.8) definierte Element kann bei XML-basierten Signaturen gemäß [RFC3275] die Platzierung der Signatur im Dokument angegeben werden. Bei anderen Signaturtypen wird das Element ignoriert und eine Warnung (Fehlercode 4197, Parameter SignaturePlacement wurde ignoriert) zurückgeliefert. |
| | `dss:ReturnUpdatedSignature` | Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene XML- oder CMS-Signatur mit zusätzlichen Informationen und Signaturen (Parallel- und Gegensignaturen) versehen werden. Hierbei sind folgende Ausprägungen für das `Type`-Attribut vorgesehen:<br>• http://ws.gematik.de/conn/sig/sigupdate/parallel<br>Hierdurch wird eine Parallelsignatur zu einer bereits existierenden Signatur erzeugt und entsprechend zurückgeliefert.<br>• http://ws.gematik.de/conn/sig/sigupdate/counter/documentincluding<br>Hierdurch wird eine dokumentinkludierende Gegensignatur für das Dokument und alle vorhandenen parallelen Signaturen erzeugt.<br>• http://ws.gematik.de/conn/sig/sigupdate/counter/documentexcluding<br>Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt.<br>Bei anderen `Type`-Attributen wird der Fehler 4111 (Ungültiger Signaturtyp oder Signaturvariante) zurückgeliefert. |
| | `dss:Schemas` | Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schemata übergeben werden, die zur Validierung der übergebenen XML-Dokumente verwendet werden können. |
| |  | |

| | | |
|---|---|---|
| | `dss:Schema` | Dieses Element enthält ein XML-Schema zur Validierung des übergebenen XML-Dokuments. Das Attribut `RefURI` ist verpflichtend. Es kennzeichnet dabei den Namensraum des XML-Schemas entsprechend [OASIS-DSS] (Abschnitt 2.8.5) |
| | `sp:GenerateUnderSignaturePolicy` |  <br>Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die erforderliche Singnaturrichlinie ausgewählt.<br>Die im Element `sp:SignaturePolicyIdentifier` übergebene URI identifiziert die Signaturrichtlinie. |
| | `SIG:ViewerInfo` | Enthält optional die vom Konnektor in die Signatur einzubeziehende Referenzen für die Stylesheets zur Anzeige. |
| **Rückgabe** |  | |
| | `SIG:SignResponse` | Eine `SignResponse` kapselt den ausgeführten Signaturauftrag pro Dokument. Die Zuordnung zwischen `SignRequest` und `SignResponse` erfolgt über die RequestID. |
| | `CONN:Status` | Enthält den Status der ausgeführten Operation pro SignRequest. |
| | `SIG:OptionalOutputs` | Enthält (angelehnt an `dss:OptionalOutputs`) optionale Ausgangsparameter:<br>  |

| | | |
|---|---|---|
| | `SIG:DocumentWithSignature` |  <br>Pro `SignResponse` wird ein Element `SIG:DocumentWithSignature` gemäß [OASIS-DSS] (Abschnitt 3.5.8) zurückgeliefert, in dem das Dokument mit Signatur enthalten ist. Dabei werden die XML-Attribute des Elements `SIG:Document` auf dem zugehörigen `SignRequest` übernommen.<br>Ist die Signatur nicht im Dokument enthalten, wird ein leeres Element `Base64XML` oder `Base64Data` zurückgegeben. Die Signatur wird dann im Element `dss:SignatureObject` abgelegt.<br>Wenn die Signatur im Dokument enthalten ist, wird das signierte Dokument im Feld `Base64XML` bzw. `Base64Data` zurückgeliefert. In diesem Fall MUSS die `dss:SignaturePtr`-Alternative in `dss:SignatureObject` (vgl. [OASIS-DSS] Abschnitt 2.5) dazu genutzt werden, auf die in den Dokumenten enthaltenen Signaturen zu verweisen. |
| | `vr:VerificationReport` | Vom Konnektor nicht befüllt. |
| | `dss:SignatureObject` | Enthält im Erfolgsfall die erzeugte Signatur pro `SignRequest` in Form eines `dss:SignatureObject`-Elementes gemäß [OASIS-DSS] (Abschnitt 3.2). |
| **Vorbe-dingungen** | Keine | |
| **Nachbe-dingungen** | Keine | |

4.1.9.4.1 TUC_KON_037 ”Zertifikat prüfen”

⌦ **TIP1-A_4696 TUC_KON_037 „Zertifikat prüfen“**

Der Konnektor MUSS den technischen Use Case „Zertifikat prüfen“ gemäß TUC_KON_037 „Zertifikat prüfen“ umsetzen.⌫

**Tabelle 6: TAB_KON_769 TUC_KON_037 „Zertifikat prüfen“**

| Element | Beschreibung |
|---|---|
| Name | TUC_KON_037 „Zertifikat prüfen“ |
| Beschreibung | Der TUC beschreibt<br>• die Prüfung eines X.509-Zertifikats gegen den Vertrauensraum |
| Auslöser | • Aufruf in einem Fachmodul oder<br>• technischen Use Case |
| Vorbedingungen | • aktuelle TSL-Informationen im Truststore vorhanden<br>• für QES X.509-Prüfung: eine aktuell gültige BNetzA-VL |
| Eingangsdaten | • certificate<br>(ein X.509-Zertifikat (nonQES- oder QES-X.509-Zertifikat))<br>• qualifiedCheck [not_required \| required \| if_QC_present] –<br>(Art der Zertifikatsprüfung)<br>• baseTime – *optional/verpflichtend, wenn ein Zeitpunkt zur Prüfung vorgegeben werden soll; default: Verwendung der Systemzeit des Konnektors*<br>(Referenzzeitpunkt: Zeitpunkt, für den das Zertifikat geprüft werden soll)<br>• offlineAllowNoCheck [Boolean] – *optional; default: false*<br>(Angabe, ob im Offlinefall der Check entfallen darf.)<br>• –intendedKeyUsage – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist; wird bei QES nicht ausgewertet*<br>(Vorgesehene KeyUsage)<br>• nur für nonQES-Zertifikate:<br>o policyList<br>(Liste der zugelassenen Zertifikatstyp-OIDs gemäß [gemSpec_OID#GS-A_4445])<br>o intendedExtendedKeyUsage – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist; wird bei QES nicht ausgewertet*<br>(Vorgesehene ExtendedKeyUsage)<br>o gracePeriod – *optional/nur für nonQES-X.509-Zertifikat und wenn vom Standard abgewichen werden soll; wird bei QES nicht ausgewertet*; *default*: *CERT_OCSP_DEFAULT_GRACE_PERIOD_NONQES*<br>(OCSP-GracePeriod: maximal zulässiger Zeitraum, den letzte OCSP-Antwort aus dem Cache bezüglich des Referenzzeitpunkts zurückliegen darf;)<br>o validationMode [OCSP \| CRL \| NONE] – *optional/verpflichtend, wenn certificate ein nonQES-X.509-Zertifikat ist*<br>(Prüfmodus:<br>▪ OCSP: Es wird mittels OCSP geprüft. Dabei wird, falls die OCSP-GracePeriod noch nicht abgelaufen ist, die OCSP-Antwort aus dem Cache des Konnektors verwendet. Für QES einzig erlaubter validationMode.<br>▪ CRL: Es wird gegen die aktuelle CRL auf dem Konnektor geprüft.<br>▪ NONE: Keine Prüfung von Statusinformationen) |

| Element | Beschreibung |
|---|---|
| | • ocspResponse – *optional*<br>(OCSPResponse des EE-Zertifikats)<br>• getOCSPResponses [Boolean] – *optional; default: false*<br>(true – OCSPResponse des geprüften Zertifikats soll an den Aufrufer zurückgegeben werden)<br>~~• attribCertificates – *optional*~~<br>~~(Liste von Attributzertifikaten)~~ |
| Komponenten | Konnektor |
| Ausgangsdaten | • Status und Liste von Warnungen/Fehlern bei der Zertifikatsprüfung<br>• role<br>(aus dem Zertifikate ermittelte Rolle oder Berufsgruppe; siehe „Tab_PKI_406 OID-Festlegung technische Rolle in X.509-Zertifikaten“ oder „Tab_PKI_402 OID-Festlegung Rolle im X.509-Zertifikat für Berufsgruppen“ oder Tab_PKI_403 OID-Festlegung Institutionen im X.509-Zertifikat der SMC-B [gemSpec_OID])<br>• qcStatement – *optional/verpflichtend, wenn certificate ein QES-X.509-Zertifikat ist*,<br>(QCStatements des Zertifikats)<br>• ocspResponsesRenewed – *optional/verpflichtend, wenn Eingabeparameter getOCSPResponses = true*<br>(OCSP-Response des geprüften Zertifikats) |
| Standardablauf | 1. Wenn das X.509-Zertifikat von einem CA-Zertifikat ausgestellt wurde, das in CERT_IMPORTED_CA_LIST enthalten ist, erfolgt eine Zertifikatsprüfung analog zu den Festlegungen in TUC_PKI_018 „Zertifikatsprüfung”.<br>Dabei sind zu prüfen:<br>- Zeitliche Gültigkeit,<br>- mathematische Prüfung der Zertifikatssignatur,<br>- die Prüfung der Zweckbindung gemäß der im Zertifikat hinterlegten keyUsage.<br>TSL-bezogene Prüfungen im TUC_PKI_018 werden in diesem Fall nicht durchgeführt. Ebenso erfolgt keine OCSP-Prüfung.<br><br>2. Wenn das zum X.509-Zertifikat gehörende CA-Zertifikat nicht in CERT_IMPORTED_CA_LIST enthalten ist, werden, abhängig vom Parameter *qualifiedCheck* folgende TUCs unter Weitergabe aller Eingangsparameter sowie der Negation des Werts von MGM_LU_ONLINE als Parameter „Offline-Modus“ aufgerufen:<br>i. Für *qualifiedCheck* = not_required: TUC_PKI_018 „Zertifikatsprüfung in der TI”<br>Ist der Eingangsparameter ocspResponses mit einer OCSP-Antwort gefüllt, so wird dieser übergeben.<br>Die aktuell aus der OCSP-Abfrage resultierte OCSP-Antwort, falls vorhanden, wird an den Aufrufer weitergegeben.<br>ii. Für *qualifiedCheck* = required: TUC_PKI_030 „QES-Zertifikatsprüfung“<br>Dabei wird ~~werden~~ das Basiszertifikat ~~und die Attributzertifikate~~ übergeben.<br>Ist Eingangsparamter ocspResponses mit einer OCSP-Response gefüllt, so wird dieser übergeben. |

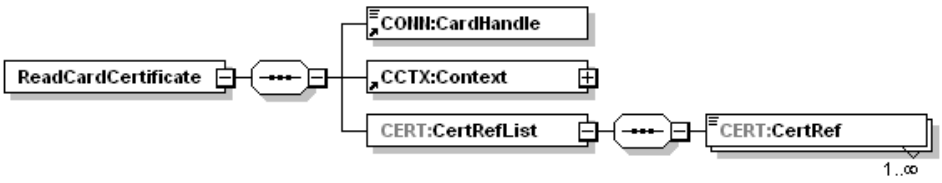
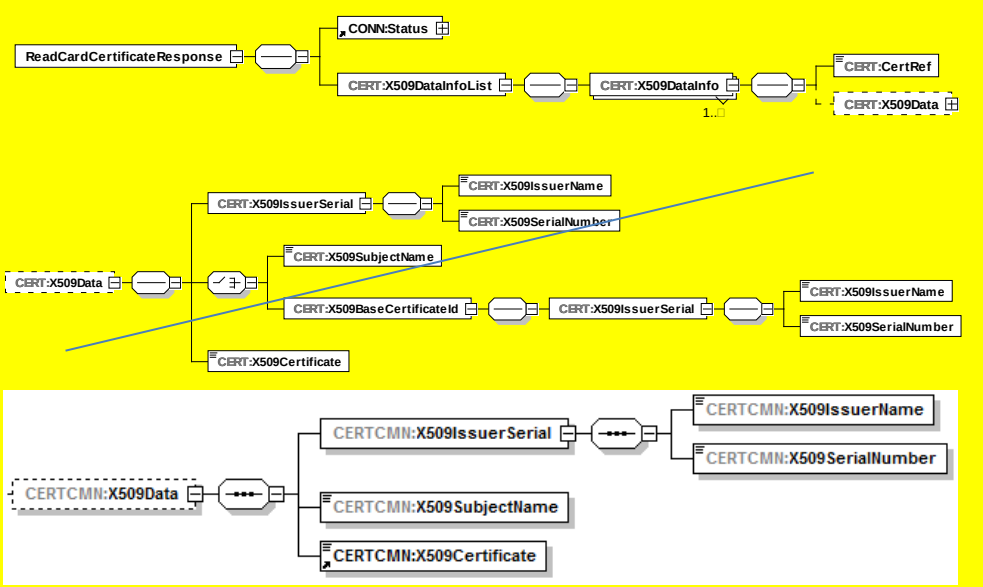
| Element | Beschreibung |
|---|---|
| | Die aktuell aus der OCSP-Abfrage resultierende OCSP-Response, falls vorhanden, wird an den Aufrufer weitergegeben.<br>iii. Für *qualifiedCheck* = if_QC_present: Ist im jeweiligen Signaturzertifikat mindestens ein QCStatement mit dem OID id-etsi-qcs-QcCompliance (0.4.0.1862.1.1) enthalten, handelt es sich um eine QES-Zertifikatsprüfung mittels TUC_PKI_030 „QES-Zertifikatsprüfung“, sonst um eine nonQES-Zertifikatsprüfung mittels TUC_PKI_018 „Zertifikatsprüfung“<br>Als Timeout wird beim Aufruf von TUC_PKI_018 der Wert von CERT_OCSP_TIMEOUT_NONQES bzw. beim Aufruf von TUC_PKI_030 der Wert von CERT_OCSP_TIMEOUT_QES übergeben (siehe auch Eingangsdaten von diesen TUCs in [gemSpec_PKI]).<br>Für die QES-Zertifikatsprüfung wird das zu prüfende QES-Zertifikat an TUC_PKI_030 „QES-Zertifikatsprüfung“ übergeben.<br>Wird im Aufruf der Eingangsparameter getOCSPResponses = false mit übergeben, wird keine OCSP-Response an den Aufrufer zurückgegeben.<br>Wenn der Eingangsparameter validationMode („Prüfmodus“) den Wert NONE hat, werden die TUC_PKI_018-Eingangsparameter<br>• „Offline-Modus“ unabhängig von MGM_LU_ONLINE auf „ja“ gesetzt und<br>• „Prüfmodus“ auf „OCSP“.<br>3. Der Status der Prüfung wird zurückgegeben. |
| Varianten/Alternativen | |
| Fehlerfälle | Keine außer denen der aufgerufenen TUCs |
| Nichtfunktionale Anforderungen | Der Konnektor MUSS unter Einhaltung aller anderen Anforderungen an die Zertifikatsprüfung die Anzahl der OCSP-Abfragen minimieren. Dies MUSS durch Caching (unter Berücksichtigung der Grace Period) und DARF NICHT durch Bündelung von OCSP-Anfragen geschehen. |
| Zugehörige Diagramme | keine |

#### 4.1.9.5.2 ReadCardCertificate

**⌦ TIP1-A_4700 Operation ReadCardCertificate**

Die Basisanwendung Zertifikatsdienst des Konnektors MUSS an der Client-schnittstelle eine Operation ReadCardCertificate wie in Tabelle 7: TAB_KON_678 Operation ReadCardCertificate beschrieben anbieten**.**

**Tabelle 7: TAB_KON_678 Operation ReadCardCertificate**

| | | |
|---|---|---|
| **Name** | `ReadCardCertificate` | |
| **Beschreibung** | Liest X.509-Zertifikate von einer Karte. | |
| **Aufruf-parameter** |  | |
| | **Name** | **Beschreibung** |
| | `CardHandle` | Gibt die Karte an, von der das Zertifikat gelesen werden soll.<br>Es können Zertifikate von HBAx (HBA, HBA-VK), SM-B ausgelesen werden.<br>Die Operation ReadCardCertificate DARF das Lesen von Zertifikaten der eGK NICHT unterstützen. |
| | `Context` | Aufrufkontext (Mandant) |
| | `CertRefList` | Gibt an, welche(s) Zertifikat(e) gelesen werden soll. Mögliche Werte für CertRef sind:<br><br>`C.AUT, C.ENC, C.SIG, C.QES,` ~~`C.QES-AC1, C.QES-AC2, C.QES-AC3`~~ |
| **Rückgabe** |  | |
| | `Status` | Enthält den Ausführungsstatus der Operation. |
| | `CertRef` | Dieses Element beinhaltet die Referenz des Zertifikats, welches bei der Anfrage übergeben wurde. |
| | `X509Data` | Inhalt des über die CertRef referenzierten Zertifikats. Ist das referenzierte Zertifikat nicht vorhanden, so wird dieses Element nicht vom Konnektor gefüllt. |
| | | `X509IssuerN` Enthält den Issuer-Name des Zertifikats. |

| | | | |
|---|---|---|---|
| | | ame | Bezüglich des Encodings sind die in [XMLDSig#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.) |
| | | X509SerialNumber | Enthält die serialNumber des Zertifikats. |
| | | X509SubjectName | Enthält ~~im Falle eines Public-Key-Zertifikats~~ das Feld subject.CommonName. Bezüglich des Encodings sind die in [XML DSig#4.4.4.4.1] angegebenen Regeln zu beachten (Escaping von Sonderzeichen etc.) |
| | | ~~X509BaseCertificateId~~ | ~~Enthält im Falle eines Attributzertifikats Issuer-Name und Seriennummer des Basiszertifikates, falls vorhanden.~~ |
| | | X509Certificate | Enthält das base64-codierte Zertifikat, dessen Binärstruktur wiederum ASN.1-codiert (gemäß [COMMON_PKI]) vorliegt. |
| **Vorbe-dingungen** | Keine | | |
| **Nachbe-dingungen** | Keine | | |

Der Ablauf der Operation ReadCardCertificate ist in Tabelle 8: TAB_KON_679 Ablauf ReadCardCertificate beschrieben:

**Tabelle 8: TAB_KON_679 Ablauf ReadCardCertificate**

| Nr. | Aufruf Technischer Use Case oder Interne Operation | Beschreibung |
|---|---|---|
| | checkArguments | Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.<br>Wurde als Zielkarte eine eGK adressiert, wird Fehlercode 4090 zurückgeliefert. |
| 2. | TUC_KON_000 „Prüfe Zugriffs-berechtigung“ | Die Prüfung erfolgt durch den Aufruf<br>TUC_KON_000 {<br>mandantId = $context.mandantId;<br>clientsystemId = $context.clientsystemId;<br>workplaceId = $context.workplaceId;<br>userId = $context.userId;<br>cardHandle = $cardHandle<br>}<br>Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab. |
| 3. | TUC_KON_026 „Liefere CardSession“ | Ermittle CardSession über TUC_KON_026 {<br>mandatId =$context.mandantId;<br>clientsystemId = $context.clientsystemId; |

| Nr. | Aufruf Technischer Use Case oder Interne Operation | Beschreibung |
|---|---|---|
| | | cardHandle = $context.cardHandle;<br>userId = $context.userId } |
| 4. | getEF | Für jedes Paar von CertRef und CardHandle wird gemäß folgender Tabelle das zu lesende File (EF) bestimmt:<br>Referenz EF<br>C.AUT HBA-VK: EF.C.HP.AUT<br>HBA: EF.C.HP.AUT.R2048<br>SM-B: EF.C.HCI.AUT<br>C.ENC HBA-VK: EF.C.HP.ENC<br>HBA: EF.C.HP.ENC.2048<br>SM-B: EF.C.HCI.ENC.R2048<br>C.SIG SM-B: EF.C.HCI.OSIG.R2048<br>C.QES HBA-VK: EF.C.HP.QES<br>HBA: EF.C.HP.QES.R2048<br>~~C.QES-AC1 HBAx: EF.C.HP.QES-AC1~~<br>~~C.QES-AC2 HBAx: EF.C.HP.QES-AC2~~<br>~~C.QES-AC3 HBAx: EF.C.HP.QES-AC3~~<br>Ist die übergebene Zertifikatsreferenz ungültig, wird Fehlercode 4149 zurückgegeben.<br>Das Lesen von Zertifikaten der eGK ist aus Sicherheitsgründen für Clientsysteme nicht zulässig. |
| 5. | TUC_KON_216 „LeseZertifikat“ | Für jedes Paar von CardHandle und EF wird nun durch Aufruf von TUC_KON_216 „LeseZertifikat“ das Zertifikat ausgelesen. |
| 6. | Zertifikatsattribute extrahieren | Aus jedem Zertifikat werden die zu liefernden Attribute extrahiert. Die Ergebnisstruktur wird mit den erhaltenen Rückgabewerten gefüllt. |

**Tabelle 9: TAB_KON_604 Fehlercodes „ReadCardCertificate“**

| Fehlercode | ErrorType | Severity | Fehlertext |
|---|---|---|---|
| Neben den Fehlercodes der aufgerufenen technischen Use Cases können folgende weitere Fehlercodes auftreten: | | | |
| 4000 | Technical | Error | Syntaxfehler |
| 4149 | Technical | Error | Ungültige Zertifikatsreferenz |
| 4090 | Security | Error | Zugriff auf eGK nicht gestattet |

⌫

## B3 – Profilierung VerificationReport

~~**Signatur mit Attributzertifikaten**~~

~~Es wird für jedes Attributzertifikat, das von der Signatur umfasst wird, ein Element~~

~~/VerificationReport/~~
~~IndividualReport/~~
~~SignedObjectIdentifier/~~
~~SignedProperties/~~
~~SignedSignatureProperties/~~
~~SignerRole/~~
~~CertifiedRoles/~~
~~AttributeCertificateValidity~~

~~angelegt.~~

## Anhang D - Übersicht über die verwendeten Versionen

| | |
|---|---|
| XSD Name | SignatureService.xsd |
| XSD Schemaversion | ~~7.4.1~~ 7.4.2 |
| TargetNamespace | http://ws.gematik.de/conn/SignatureService/v7.4 |

| | |
|---|---|
| XSD Name | CertificateServiceCommon.xsd |
| XSD Schemaversion | ~~2.0.0~~ 2.0.1 |
| TargetNamespace | http://ws.gematik.de/conn/CertificateServiceCommon/2.0 |

SignatureService.xsd

```
<schema ... version="~~7.4.1~~7.4.2">
  …
<element name="SignRequest">
      <complexType>
         <sequence>
            <element name="OptionalInputs" minOccurs="0">
               <complexType>
                  <all>
                     <element ref="dss:SignatureType" minOccurs="0"/>
                     <element ref="dss:Properties" minOccurs="0"/>
                     ~~<element name="AttributeCertificates" minOccurs="0">~~
                        ~~<complexType>~~
                           ~~<choice>~~
                              ~~<element ref="CERTCMN:X509DataInfoList"/>~~
                              ~~<element name="IncludeFromCard" type="boolean"/>~~
                           ~~</choice>~~
                        ~~</complexType>~~
                     ~~</element>~~
```

```
                <element ref="SIG:IncludeEContent" minOccurs="0"/>
                <element name="IncludeObjects" minOccurs="0">
                  <complexType>
                    <sequence>
                      <element ref="dss:IncludeObject" maxOccurs="unbounded"/>
                    </sequence>
                  </complexType>
                </element>
                <element ref="dss:SignaturePlacement" minOccurs="0"/>
                <element ref="dss:ReturnUpdatedSignature" minOccurs="0"/>
                <element ref="dss:Schemas" minOccurs="0"/>
                <element ref="sp:GenerateUnderSignaturePolicy" minOccurs="0"/>
                <element ref="SIG:ViewerInfo" minOccurs="0"/>
              </all>
            </complexType>
          </element>
          <element ref="SIG:Document"/>
          <element ref="SIG:IncludeRevocationInfo"/>
        </sequence>
        <attribute name="RequestID" type="string" use="required"/>
      </complexType>
    </element>
```

CertificateServiceCommon.xsd

```
<schema … version="2.0.02.0.1"    >

  <simpleType name="CertRefEnum">
    <restriction base="token">
      <enumeration value="C.AUT"/>
      <enumeration value="C.ENC"/>
      <enumeration value="C.SIG"/>
      <enumeration value="C.QES"/>
      <enumeration value="C.QES-AC1"/>
      <enumeration value="C.QES-AC2"/>
      <enumeration value="C.QES-AC3"/>
    </restriction>
  </simpleType>

  <complexType name="X509DataInfoListType">
    <sequence>
      <element name="X509DataInfo" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="CertRef" type="CERTCMN:CertRefEnum"/>
            <element name="X509Data" minOccurs="0">
              <complexType>
                <sequence>
                  <element name="X509IssuerSerial">
                    <complexType>
                      <sequence>
                        <element name="X509IssuerName" type="string"/>
                        <element name="X509SerialNumber" type="string"/>
                      </sequence>
                    </complexType>
```

```
                    </element>
                    <choice>
                      <element name="X509SubjectName" type="string"/>
                      <element name="X509BaseCertificateId">
                        <complexType>
                          <sequence>
                            <element name="X509IssuerSerial">
                              <complexType>
                                <sequence>
                                  <element name="X509IssuerName" type="string"/>
                                  <element name="X509SerialNumber" type="string"/>
                                </sequence>
                              </complexType>
                            </element>
                          </sequence>
                        </complexType>
                      </element>
                    </choice>
                    <element ref="CERTCMN:X509Certificate"/>
                  </sequence>
                </complexType>
              </element>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
```

# Änderungen in gemSpec_HBA_ObjSys_G2.1

Die Attributzertifikate C.HP.QES-AC1, C.HP.QES-AC2 und C.HP.QES-AC3 in den Ausprägungen R2048 und E256 werden durch die Telematikinfrastruktur nicht mehr unterstützt.

Die vorhandene Beschreibung der Objekte zur Ablage dieser Attributzertifikate auf dem HBA der Generation G2.1 soll aus der Spezifikation des Objektssystems des HBA G2.1 entfernt werden.

Die Spezifikation gemSpec_HBA_ObjSys_G2.1 soll wie folgt geändert werden:

## 1. Entfernen HBA-Objekte für Attributzertifikate RSA

Das Kapitel ”5.5.2.5MF / DF.QES / EF.C.HP.QES-AC1, MF / DF.QES / EF.C.HP.QES-AC2 und MF / DF.QES / EF.C.HP.QES-AC3” und alle enthaltenen Unterkapitel werden vollständig entfernt.

### ~~5.5.2.5 MF / DF.QES / EF.C.HP.QES-AC1, MF / DF.QES / EF.C.HP.QES-AC2 und MF / DF.QES / EF.C.HP.QES-AC3~~

## 2. Entfernen HBA-Objekte für Attributzertifikate ELC

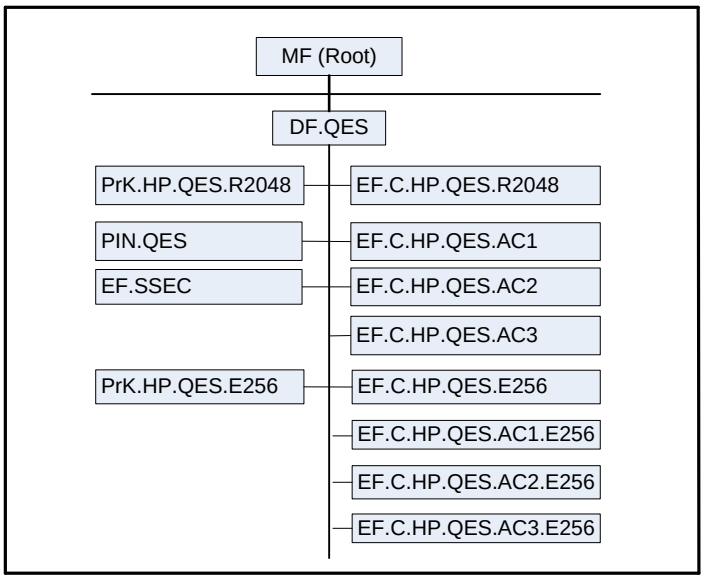
Das Kapitel " 5.5.2.8 MF / DF.QES / EF.C.HP.QES-AC1.E256, MF / DF.QES / EF.C.HP.QES-AC2.E256 und MF / DF.QES / EF.C.HP.QES-AC3.E256" und alle enthaltenen Unterkapitel werden vollständig entfernt

~~5.5.2.8 MF / DF.QES / EF.C.HP.QES-AC1.E256, MF / DF.QES / EF.C.HP.QES-AC2.E256 und MF / DF.QES / EF.C.HP.QES-AC3.E256~~

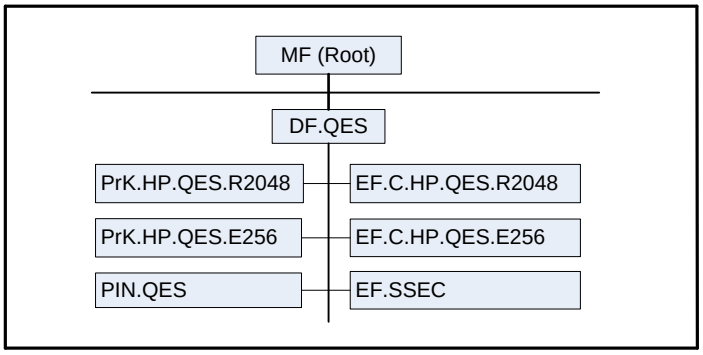
## 3. Anpassung der Strukturgrafik DF.QES

In Kapitel 5.5.1 "Dateistruktur und Dateiinhalt" wird die Abbildung Abb_HBA_ObjSys_003 wie folgt geändert.

Bisher:




Neu:



## 4. Verweis auf Attributzertifikate im Text entfernen

In Kapitel 5.5.1 wird die Beschreibung wie folgt geändert:

Die QES-Anwendung beinhaltet~~besitzt~~ EFs für die~~das~~ X.509-QES-Zertifikate ~~und maximal drei Attributzertifikate, jeweils~~ für die Kryptographie mit RSA und mit elliptischen Kurven. Zusätzlich ist ein EF zur Anzeige des unterstützten Maximalwertes des SSEC angelegt.

## 5. Anpassung Größe und Inhalt EF.CD der Cryptografic Information Application (CIA) für die Anwendung QES

In Kapitel 5.7.1.5 MF/DF.CIA.QES/EF.CD (Certificate Directory) wird der Inhalt von Tab_HBA_ObjSys_063 wie folgt geändert

**Tabelle 92: Tab_HBA_ObjSys_063 Initialisierte Attribute von MF/ DF.CIA.QES / EF.CD (Certificate Directory)**

| Attribute | Wert | Bemerkung |
|---|---|---|
| Objekttyp | transparentes Elementary File | |
| *fileIdentifier* | ’50 38’ | |
| *shortFileIdentifier* | ‘16’ = 22 | |
| *numberOfOctet* | ~~’01 50’ Oktett = 336 Oktett~~<br>’4E’ Oktett = 78 Oktett | |
| *positionLogicalEndOfFile* | ~~’01 50’ Oktett = 336 Oktett~~<br>’4E’ Oktett = 78 Oktett | |
| *flagTransactionMode* | False | |
| *flagChecksum* | True | |
| *lifeCycleStatus* | Operational state (activated) | |
| *shareable* | True | |
| *body* | 30 25<br>30 0A<br>0C 08 43 2E 48 50 2E 51 45 53<br>30 03<br>04 01 84<br>A1 12<br>30 10<br>30 0E<br>A1 0C<br>4F 06 D2 76 00 00 66 01<br>04 02 C0 00<br>~~A0 29~~<br>~~30 0E~~<br>~~0C 0C 43 2E 48 50 2E 51 45 53~~<br>~~2D 41 43 31~~<br>~~30 03~~<br>~~04 01 84~~<br>~~A1 12~~ | |

```
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 01
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
            2D 41 43 32
   30 03
      04 01 84
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 02
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
            2D 41 43 33
   30 03
      04 01 84
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 03
30 25
   30 0A
      0C 08 43 2E 48 50 2E 51 45 53
   30 03
      04 01 86
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 06
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
            2D 41 43 31
   30 03
      04 01 86
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 07
A0 29
   30 0E
      0C 0C 43 2E 48 50 2E 51 45 53
            2D 41 43 32
   30 03
      04 01 86
   A1 12
      30 10
         30 0E
            A1 0C
               4F 06 D2 76 00 00 66 01
               04 02 C0 08
A0 29
   30 0E
```

| | | |
|---|---|---|
| | ~~0C 0C 43 2E 48 50 2E 51 45 53~~<br>~~2D 41 43 33~~<br>~~30 03~~<br>~~04 01 86~~<br>~~A1 12~~<br>~~30 10~~<br>~~30 0E~~<br>~~A1 0C~~<br>~~4F 06 D2 76 00 00 66 01~~<br>~~04 02 C0 09~~ | |

### Hinweis

Durch die Änderungen in gemSpec_HBA_ObjSys_G2.1 entfallen die Anforderungen:

Card-G2-A_2094-01
Card-G2-A_3302
Card-G2-A_2095-01
Card-G2-A_3303
Card-G2-A_2096-01
Card-G2-A_3304
Card-G2-A_3633
Card-G2-A_3634
Card-G2-A_3635
Card-G2-A_3636
Card-G2-A_3637
Card-G2-A_3638

Durch die Änderungen in gemSpec_HBA_ObjSys_G2.1 wird folgende Anforderung inhaltlich geändert:

Card-G2-A_2123-01

Die dargestellten Änderungen in gemSpec_HBA_ObjSys_G2.1 ergeben Veränderungen in den folgenden Produkttypsteckbriefen.

# Änderungen in gemProdT_HBA_G2.1

**Tabelle 10: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test"**

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| GS-A_5020 | Einbringung des Komponentenzertifikats durch den | gemRL_TSL_SP_CP |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| | Kartenherausgeber | |
| TIP1-A_2578 | Korrekte ICCSN der Chipkarte | gemSpec_CVC_TSP |
| TIP1-A_2589 | Personalisierung des CVC-CA-Zertifikats | gemSpec_CVC_TSP |
| Card-G2-A_3673 | K_Personalisierung und K_Initialisierung:Unterstützung RSA CV-Zertifikate | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3009 | K_HBA: Zusatzanforderungen für kontaktlose Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2038 | K_Personalisierung: Druck der CAN auf den HBA bei Verwendung der optionalen kontaktlosen Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2673 | K_Personalisierung und K_Initialisierung: Wert von „positionLogicalEndOfFile“ | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3005 | Absicherung der Kartenadministration | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2040-01 | K_Personalisierung und K_Initialisierung: Wert des Attributes answerToReset | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2041 | K_Personalisierung: Wert des Attributes iccsn8 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3395 | K_Personalisierung: personalisierter Wert von pointInTime | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2043 | K_Personalisierung und K_Initialisierung: ATR-Kodierung | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2044-01 | K_Personalisierung und K_Initialisierung: TC1 Byte im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3627 | K_Personalisierung und K_Initialisierung: T0 Byte im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3015 | K_Personalisierung und K_Initialisierung: Historical Bytes im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2045 | K_Personalisierung und K_Initialisierung: Vorgaben für Historical Bytes | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2046 | K_Personalisierung: Kompatibilität zu G1-Karten | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2058-01 | K_Personalisierung: Personalisiertes Attribut von EF.GDO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3281 | K_Personalisierung: Personalisierte Attribute von MF / EF.C.CA_HPC.CS.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3282 | K_Personalisierung: Personalisierte Attribute von MF / EF.C.CA_HPC.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3385 | K_Personalisierung: Festlegung von CHR in MF / EF.C.HPC.AUTR_CVC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3283 | K_Personalisierung: Personalisierte Attribute von MF / EF.C.HPC.AUTR_CVC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3386 | K_Personalisierung: Festlegung von CHR in MF / EF.C.HPC.AUTR_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3284 | K_Personalisierung: Personalisierte Attribute von MF / EF.C.HPC.AUTR_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3387 | K_Personalisierung: Festlegung von CHR in MF / EF.C.HPC.AUTD_SUK_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3285 | K_Personalisierung: Personalisierte Attribute von MF / EF.C.HPC.AUTD_SUK_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3326 | K_Personalisierung: Personalisierte Attribute von MF / PuK.RCA.CS.R2048 für Testkarten | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3327-01 | K_Personalisierung: Personalisierte Attribute von MF / PuK.RCA.CS.E256 für Testkarten | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3297 | K_Personalisierung: Personalisierte Attribute von MF / SK.CAN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2869 | K_Personalisierung: Generierung der CAN bei Verwendung der optionalen kontaktlosen Schnittstelle des HBA | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3301 | K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| ~~Card-G2-A_3302~~ | ~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3303~~ | ~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3304~~ | ~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| Card-G2-A_3630 | K_Personalisierung: Personalisierte Attribute von MF / DF.QES / PrK.HP.QES.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3632 | K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES.E256 | gemSpec_HBA_ObjSys_G2.1 |
| ~~Card-G2-A_3634~~ | ~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3636~~ | ~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3638~~ | ~~K_Personalisierung: Personalisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| Card-G2-A_3307 | K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3308 | K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3640 | K_Personalisierung: Personalisierte MF / DF.ESIGN / PrK.HP.AUT.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3642 | K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3644 | K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3646 | K_Personalisierung: Personalisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3317-01 | K_Personalisierung: Personalisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO1.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3318-01 | K_Personalisierung: Personalisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO2.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3479 | Kodierung von Versionskennungen | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3480 | Kodierung von Produktidentifikatoren | gemSpec_Karten_Fach_TIP_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3481 | Ausschluss für die Kodierung von Produktidentifikatoren | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3487 | K_Initialisierung und K_Personalisierung: DO_HistoricalBytes in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3492 | K_Personalisierung: DO_PT_Pers in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3494 | K_Personalisierung: DO_PI_Kartenkörper in EF.ATR-Personalisierung | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3495 | K_Personalisierung: DO_PI_Personalisierung in EF.ATR-Personalisierung | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3496 | K_Initialisierung: Weitere Datenobjekte in DO_HistoricalBytes in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3497 | K_Personalisierung: Vollständige Befüllung von EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3498 | K_Personalisierung: DO_ICCSN in EF.GDO | gemSpec_Karten_Fach_TIP_G2.1 |
| GS-A_4377 | Card-to-Card-Authentisierung G1 | gemSpec_Krypt |
| GS-A_3695 | Grundlegender Aufbau Versionsnummern | gemSpec_OM |
| GS-A_5026 | Versionierung von Karten durch die Produktidentifikation | gemSpec_OM |
| GS-A_5140 | Inhalt der Selbstauskunft von Karten | gemSpec_OM |
| GS-A_4559 | Versionierung der Karten der TI | gemSpec_OM |
| GS-A_4560 | Versionierung von Datenstrukturen der Karten der TI | gemSpec_OM |
| GS-A_4583 | Berufsgruppenkennzeichen für HBA | gemSpec_PKI |
| GS-A_4587 | Gesamtlänge der Telematik-ID | gemSpec_PKI |
| GS-A_4710 | Präfix der Telematik-ID | gemSpec_PKI |
| GS-A_4711 | Separator der Telematik-ID | gemSpec_PKI |
| GS-A_4974 | CV-Ausstattung von Smartcards der TI | gemSpec_PKI |
| GS-A_4621 | Zugriffsprofil von HBA und SM-B (SMC-B, HSM-B) | gemSpec_PKI |
| GS-A_4623 | Zugriffsprofil eines HBA | gemSpec_PKI |

# Änderungen in gemProdT_HBA_ObjSys_G2.1

**Tabelle 11: Anforderungen zur funktionalen Eignung "Produkttest / Produktübergreifender Test"**

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_2232 | Größe Chip-Modul | gemSpec_eGK_OPT |
| Card-G2-A_2036 | K_Initialisierung Anzahl logischer Kanäle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3006 | K_HBA: USB-Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2867 | K_HBA: Vorhandensein einer USB-Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3007 | K_HBA: Kontaktlose Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3009 | K_HBA: Zusatzanforderungen für kontaktlose Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3010 | K_Initialisierung und K_Personalisierung: Kontaktlose Schnittstelle wird nicht genutzt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3011 | K_Initialisierung: Kontaktlose Schnittstelle im COS nicht vorhanden | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3277 | K_Personalisierung und K_Initialisierung: Konformität kontaktlose Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3014 | K_HBA: Vorhandensein Kryptobox | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3182 | K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, kein konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3184 | K_Initialisierung: Verhalten der Objekte, konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3185 | K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2033-01 | K_Initialisierung: Ordnerattribute | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3626 | K_Initialisierung: Herstellerspezifischer FileIdentifier | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2673 | K_Personalisierung und K_Initialisierung: Wert von „positionLogicalEndOfFile“ | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2035 | K_Initialisierung: Zugriffsregeln für besondere Kommandos | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2039 | K_Initialisierung: Wert des Attributes root | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2040-01 | K_Personalisierung und K_Initialisierung: Wert des Attributes answerToReset | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2042 | K_Initialisierung: Inhalt persistentPublicKeyList | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3266-01 | K_Initialisierung: Wert von pointInTime | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2043 | K_Personalisierung und K_Initialisierung: ATR-Kodierung | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2044-01 | K_Personalisierung und K_Initialisierung: TC1 Byte im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3627 | K_Personalisierung und K_Initialisierung: T0 Byte im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3015 | K_Personalisierung und K_Initialisierung: Historical Bytes im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2045 | K_Personalisierung und K_Initialisierung: Vorgaben für Historical Bytes | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2047 | K_Initialisierung: Initialisierte Attribute von MF | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2048-01 | K_Initialisierung: Initialisierte Attribute von MF / EF.ATR | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3278 | K_Initialisierung: Initialisiertes Attribut numberOfOctet von MF / EF.ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3199 | K_Initialisierung: Initialisierte Attribute von MF / EF.CardAccess | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3628 | K_Initialisierung: Inhalt der Records von EF.DIR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2055-01 | K_Initialisierung: Initialisierte Attribute von MF / EF.DIR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2057 | K_Initialisierung: Initialisierte Attribute von MF / EF.GDO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2059-01 | K_Initialisierung: Attribute von MF / EF.Version2 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2060 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2061 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2063 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2064 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2067 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTD_SUK_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2069 | K_Initialisierung: Initialisierte Attribute von MF / PIN.CH | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2071 | K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2072 | K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2075 | K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTD_SUK_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2077-01 | K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2078-01 | K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3016-01 | K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.ADMINCMS.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2080-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES128 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2081-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3293-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES128 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3295-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2868 | K_Initialisierung: Initialisierte Attribute von MF / SK.CAN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2082 | K_Initialisierung: Initialisierte Attribute von MF / DF.HPA | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_2083 | K_Initialisierung: Initialisierte Attribute von MF / DF.HPA / EF.HPD | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2084 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2085-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2088-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PIN.QES | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2089 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.SSEC | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2090-01 | K_Initialisierung: Inhalt von EF.SSEC | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2091-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| ~~Card-G2-A_2094-01~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_2095-01~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_2096-01~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| Card-G2-A_3629-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3631 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.E256 | gemSpec_HBA_ObjSys_G2.1 |
| ~~Card-G2-A_3633~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3635~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3637~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| Card-G2-A_2097 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2098-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2101 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2107 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2110-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3639-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3641-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3643 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3645 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.E256 | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_2117 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2119-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CIA.CIAInfo | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2120-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.OD (Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2121-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.AOD (Authentication Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2122-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.PrKD (Private Key Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2123-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CD(Certificate Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2118 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3320-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CIA.CIAInfo | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3321-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.OD (Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3322-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.AOD (Authentication Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3323-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.PrKD (Private Key Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3324-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CD (Certificate Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2124 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2125-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PrK.HP.AUTO.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2128 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PIN.AUTO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2129 | K_Initialisierung: Initialisierte Attribute von MF / DF:ATUO / PIN.SO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2130-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO1.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2131-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO2.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3479 | Kodierung von Versionskennungen | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3480 | Kodierung von Produktidentifikatoren | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3481 | Ausschluss für die Kodierung von Produktidentifikatoren | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3483-01 | K_Initialisierung: Inhalt body von EF.Version2 | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3484 | K_Initialisierung: Reihenfolge der Datenobjekte in body von EF.Version2 | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3485 | K_Initialisierung: Datenobjekte in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3486 | K_Initialisierung: DO_BufferSize in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3487 | K_Initialisierung und K_Personalisierung: DO_HistoricalBytes in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3488 | K_Initialisierung: DO_PT_COS in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3489 | K_Initialisierung: DO_PI_CHIP in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3490 | K_Initialisierung: DO_PI_COS in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3491 | K_Initialisierung: DO_PI_InitialisiertesObjSys in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3493 | K_Initialisierung DO_PI_Kartenkörper in EF.ATR-Initialisierung | gemSpec_Karten_Fach_TIP |
| GS-A_4377 | Card-to-Card-Authentisierung G1 | gemSpec_Krypt |
| GS-A_3695 | Grundlegender Aufbau Versionsnummern | gemSpec_OM |
| GS-A_3700 | Versionierung von Produkten auf Basis von dezentralen Produkttypen der TI-Plattform durch die Produktidentifikation | gemSpec_OM |
| GS-A_5026 | Versionierung von Karten durch die Produktidentifikation | gemSpec_OM |
| GS-A_5054 | Versionierung von Produkten durch die Produktidentifikation erweitert um Klartextnamen | gemSpec_OM |
| GS-A_5140 | Inhalt der Selbstauskunft von Karten | gemSpec_OM |
| GS-A_4559 | Versionierung der Karten der TI | gemSpec_OM |
| GS-A_4668 | Prüfung der mathematischen Korrektheit bei CV-Zertifikaten der Generation G1 | gemSpec_PKI |

**Tabelle 12: Anforderungen zur sicherheitstechnischen Eignung "Sich.techn. Eignung: Zertifizierung nach Technischer Richtlinie"**

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_2036 | K_Initialisierung Anzahl logischer Kanäle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3009 | K_HBA: Zusatzanforderungen für kontaktlose Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3277 | K_Personalisierung und K_Initialisierung: Konformität kontaktlose Schnittstelle | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2032 | K_Initialisierung: Änderung von Zugriffsregeln | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3182 | K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, kein konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3183 | K_Initialisierung: Eigenschaften der Objekte in anderen SEs, kein konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3184 | K_Initialisierung: Verhalten der Objekte, konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3185 | K_Initialisierung: Verwendbarkeit der Objekte in anderen SEs, konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3186 | K_Initialisierung: Eigenschaften der Objekte in anderen SEs, konkretes SE genannt | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2033-01 | K_Initialisierung: Ordnerattribute | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2034 | K_Initialisierung: Dateiattribute | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2673 | K_Personalisierung und K_Initialisierung: Wert von „positionLogicalEndOfFile“ | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2035 | K_Initialisierung: Zugriffsregeln für besondere Kommandos | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3325 | K_Initialisierung und K_Personalisierung: Abweichung von Festlegungen zum Zwecke der Personalisierung | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2039 | K_Initialisierung: Wert des Attributes root | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2040-01 | K_Personalisierung und K_Initialisierung: Wert des Attributes answerToReset | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2042 | K_Initialisierung: Inhalt persistentPublicKeyList | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3181 | K_Initialisierung: Größe persistentPublicKeyList | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3266-01 | K_Initialisierung: Wert von pointInTime | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2043 | K_Personalisierung und K_Initialisierung: ATR-Kodierung | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2044-01 | K_Personalisierung und K_Initialisierung: TC1 Byte im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3015 | K_Personalisierung und K_Initialisierung: Historical Bytes im ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2045 | K_Personalisierung und K_Initialisierung: Vorgaben für Historical Bytes | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2047 | K_Initialisierung: Initialisierte Attribute von MF | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2048-01 | K_Initialisierung: Initialisierte Attribute von MF / EF.ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3278 | K_Initialisierung: Initialisiertes Attribut numberOfOctet von MF / EF.ATR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3199 | K_Initialisierung: Initialisierte Attribute von MF / EF.CardAccess | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2055-01 | K_Initialisierung: Initialisierte Attribute von MF / EF.DIR | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2057 | K_Initialisierung: Initialisierte Attribute von MF / EF.GDO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2059-01 | K_Initialisierung: Attribute von MF / EF.Version2 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2060 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2061 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.CA_HPC.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_2063 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2064 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTR_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2067 | K_Initialisierung: Initialisierte Attribute von MF / EF.C.HPC.AUTD_SUK_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2069 | K_Initialisierung: Initialisierte Attribute von MF / PIN.CH | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2071 | K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2072 | K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTR_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2075 | K_Initialisierung: Initialisierte Attribute von MF / PrK.HPC.AUTD_SUK_CVC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2077-01 | K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2078-01 | K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3016-01 | K_Initialisierung: Initialisierte Attribute von MF / PuK.RCA.ADMINCMS.CS.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2080-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES128 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2081-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CMS.AES256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3293-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES128 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3295-01 | K_Initialisierung: Initialisierte Attribute von MF / SK.CUP.AES256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2868 | K_Initialisierung: Initialisierte Attribute von MF / SK.CAN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2082 | K_Initialisierung: Initialisierte Attribute von MF / DF.HPA | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2083 | K_Initialisierung: Initialisierte Attribute von MF / DF.HPA / EF.HPD | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2084 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2085-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2088-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PIN.QES | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2089 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.SSEC | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2091-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| ~~Card-G2-A_2094-01~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_2095-01~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| ~~Card-G2-A_2096-01~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| Card-G2-A_3629-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / PrK.HP.QES.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3631 | K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES.E256 | gemSpec_HBA_ObjSys_G2.1 |
| ~~Card-G2-A_3633~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC1.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3635~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC2.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| ~~Card-G2-A_3637~~ | ~~K_Initialisierung: Initialisierte Attribute von MF / DF.QES / EF.C.HP.QES-AC3.E256~~ | ~~gemSpec_HBA_ObjSys_G2.1~~ |
| Card-G2-A_2097 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2098-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2101 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2107 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2110-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.R2048 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3639-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.AUT.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3641-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / PrK.HP.ENC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3643 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.AUT.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3645 | K_Initialisierung: Initialisierte Attribute von MF / DF.ESIGN / EF.C.HP.ENC.E256 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2117 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2119-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CIA.CIAInfo | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2120-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.OD (Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2121-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.AOD (Authentication Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2122-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.PrKD (Private Key Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2123-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.QES / EF.CD(Certificate Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2118 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3320-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CIA.CIAInfo | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3321-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.OD (Object Directory) | gemSpec_HBA_ObjSys_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3322-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.AOD (Authentication Object Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3323-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.PrKD (Private Key Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3324-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.CIA.ESIGN / EF.CD (Certificate Directory) | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2124 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2125-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PrK.HP.AUTO.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2128 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / PIN.AUTO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3270 | K_Initialisierung: CHANGE REFERENCE DATA bei Nutzung der Leer-PIN für PIN.AUTO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2129 | K_Initialisierung: Initialisierte Attribute von MF / DF:ATUO / PIN.SO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3271 | K_Initialisierung: CHANGE REFERENCE DATA bei Nutzung der Leer-PIN für PIN.SO | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2130-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO1.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2131-01 | K_Initialisierung: Initialisierte Attribute von MF / DF.AUTO / EF.C.HP.AUTO2.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_2675-01 | K_Initialisierung: Initialisierte: Wert von PrK.HP.AUTO.R3072 | gemSpec_HBA_ObjSys_G2.1 |
| Card-G2-A_3479 | Kodierung von Versionskennungen | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3480 | Kodierung von Produktidentifikatoren | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3481 | Ausschluss für die Kodierung von Produktidentifikatoren | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3483-01 | K_Initialisierung: Inhalt body von EF.Version2 | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3484 | K_Initialisierung: Reihenfolge der Datenobjekte in body von EF.Version2 | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3485 | K_Initialisierung: Datenobjekte in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3486 | K_Initialisierung: DO_BufferSize in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3487 | K_Initialisierung und K_Personalisierung: DO_HistoricalBytes in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3488 | K_Initialisierung: DO_PT_COS in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3489 | K_Initialisierung: DO_PI_CHIP in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3490 | K_Initialisierung: DO_PI_COS in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |
| Card-G2-A_3491 | K_Initialisierung: DO_PI_InitialisiertesObjSys in EF.ATR | gemSpec_Karten_Fach_TIP_G2.1 |

| Afo-ID | Afo-Bezeichnung | Quelle (Referenz) |
|---|---|---|
| Card-G2-A_3493 | K_Initialisierung DO_PI_Kartenkörper in EF.ATR-Initialisierung | gemSpec_Karten_Fach_TIP_G2.1 |
| GS-A_3695 | Grundlegender Aufbau Versionsnummern | gemSpec_OM |
| GS-A_3696 | Zeitpunkt der Erzeugung neuer Versionsnummern | gemSpec_OM |
| GS-A_3700 | Versionierung von Produkten auf Basis von dezentralen Produkttypen der TI-Plattform durch die Produktidentifikation | gemSpec_OM |
| GS-A_5026 | Versionierung von Karten durch die Produktidentifikation | gemSpec_OM |
| GS-A_5140 | Inhalt der Selbstauskunft von Karten | gemSpec_OM |
| GS-A_4559 | Versionierung der Karten der TI | gemSpec_OM |