Elektronische Gesundheitskarte und Telematikinfrastruktur

Errata 4 zum Konnektor PTV 3 (eMP/AMTS, NFDM)

Version:	1.0.1
Stand:	27.11.2019
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemErrata_4_Kon_PTV3]

betroffener Produkttyp	**neue Produkttypversion:**
gemProdT_Kon_PTV3	PTV3.6.0-0

ID	Dokument	Quelle Dokument und/oder Kapitel	Beschreibung der Änderung	**Anpassungen** an Afos; TUCs, Tabellen, Korrekturen	**von Änderung betroffene Dokumente***
C_7076	gemSpec_Kon gemSpec_Krypt	TIP1-A_4617 GS-A_4375 GS-A_4376	**RSA-Verschlüsselungsverfahren für PTV3-Konnektor** RSAES-PKCS1-v1-5 wird für den PTV3-Konnektor optional TIP1-A_4617 => TIP1-A_4617-01 GS-A_4375 => GS-A_4375-01 RSAES-OAEP wird verpflichtend GS-A_4376 => GS-A_4376-01	**gemSpec_Kon; TIP1-A_4617-01 ; neu, mit Änderungsmarkierung zu TIP1-A_4617:** **Standardablauf 1.** Das Verfahren zum Entschlüsseln wird entsprechend dem Format des übergebenen zu entschlüsselnden Dokuments (EncryptedDocument) gewählt. Der Konnektor MUSS ~~SOLL~~ beim asymmetrischen Anteil der Entschlüsselung hybrid verschlüsselter Dokumente ~~sowohl RSAES-OAEP als auch RSAES-PKCS1-v1-5 unterstützen. Er MUSS außerdem das~~ die in [gemSpec_Krypt~~#5.8~~] beschriebenen ~~ECIES~~ Verfahren unterstützen. **Diese AFO wird dem PTV3-Konnektor zugeordnet. Die AFO TIP1-A_4617 wird beim PTV3 Konnektor entfernt.** **gemSpec_Krypt; GS-A_4375-01 ; neu, mit Änderungsmarkierung zu GS-A_4375:** **XML-~~VerEnt~~schlüsselung - Hybrid, Schlüsseltransport RSAES-PKCS1-v1_5** Alle Produkttypen, die Dokumente mittels [XMLEnc-1.1] RSA-basiert hybrid ~~verent~~schlüsseln, ~~MÜSSEN~~KÖNNEN für die ~~VerEnt~~schlüsselung des symmetrischen Schlüssels den Algorithmus ~~RSAES-OAEP gemäß [PKCS#1] oder Algorithmus~~ RSAES-PKCS1-v1_5 unter Berücksichtigung von speziellen Maßnahmen gegen Seitenkanalangriffe (vgl. [BSI-TR-03116-1] S. 16) verwenden. **Diese AFO wird dem PTV3-Konnektor zugeordnet. Die AFO GS-A_4375 wird beim PTV3-Konnektor entfernt.** **gemSpec_Krypt; GS-A_4376-01 ; neu, mit Änderungsmarkierung zu GS-A_4376:** **XML-Verschlüsselung - Hybrid, Schlüsseltransport RSAES-OAEP** Alle Produkttypen, die Dokumente mittels [XMLEnc-1.1] RSA-basiert hybrid verschlüsseln, ~~SOLLEN~~MÜSSEN für den Schlüsseltransport den Algorithmus RSAES-OAEP gemäß [PKCS#1] verwenden. **Diese AFO wird dem PTV3-Konnektor zugeordnet. Die AFO GS-A_4376 wird beim PTV3-Konnektor entfernt.**	gemSpec_Kon gemSpec_Krypt gemProdT_Kon_PTV3
C_7095	gemRL_QES_NFDM	Tabelle2 / Tabelle3	**Erweiterungen und Korrekturen gemRL_QES_NFDM** In Kapitel 3.2 Typkonformität werden zusätzliche Prüfungen für den signierten Notfalldatensatz ergänzt, um zu prüfen, dass die Signatur entsprechend der Signaturrichtlinie erstellt wurde. Eine falsche URI der Signaturrichtlinie wird korrigiert.	In Kapitel 3.2 Typkonformität wird ergänzt: Bei der Verifikation wird der Typ identifiziert durch SigPolicyID/Identifyer = "urn:gematik:fa:sak:nfdm:r1:v1" in der Signatur Im Rahmen der Typenkonfomität ist darüber hinaus zu prüfen, dass • Der signierte Notfalldatensatz genau eine Signatur enthält, • Die Signatur das Element NFD:Notfalldaten referenziert. Die URI der Signaturrichtlinie wird korrigiert: - Tabelle2: /fa/nfds/NFD_Document_v1_4.xsd http://ws.gematik.de /fa/nfds/NFD_Document/v1.4 - Tabelle3: urn:gematik:fa:sak:nfdm:r1:v1 http://ws.gematik.de/fa/nfds/NFD_Document/v1.4 NFD_Document	gemRL_QES_NFDM gemProdT_Kon_PTV3 gemProdT_Kon_PTV4
C_7096	gemProT_Kon_PTV3	gemRL_QES_NFDM	**Referenz auf gemRL_QES_NFDM aktualisiert** Es wird die falsche Version von gemRL_QES_NFDM referenziert.	Tabelle 2: Mitgeltende Dokumente: [gemRL_QES_NFDM] Signaturrichtlinie QES Notfalldaten-Management (NFDM) 1.4.0	gemProdT_Kon_PTV3
C_7100	gemSpec_Kon	TIP1-A_2255-01 TIP1-A_4672-01 TIP1-A_5540-01 TIP1-A_5034-03	**Entfernen der Prüfung von Qualifizierten Zeitstempeln** Da auf absehbare Zeit in der TI keine Möglichkeit bestehen wird, qualifizierte Zeitstempel zu erstellen, wird eine Prüfung qualifizierter Zeitstempel durch den Konnektor aktuell nicht benötigt. Dabei wird auch die Inkonsistenz beseitigt, dass die Spezifikation nicht die Anforderung aus gemKPT_Arch_TIP umsetzt, da es sich um unterschiedliche Zeitstempel handelt.	Siehe C_7100_qualifizierte_Zeitstempel.docx	gemKPT_Arch_TIP gemSpec_Kon gemProdT_Kon_PTV3 gemProdT_Kon_PTV4
C_7101	gemSpec_Kon	TAB_KON_778	**Verzichtoption für nonQES-XAdES-Signatur** Um die Zertifizierung des PTV3-Konnektors zu beschleunigen, wird eine Zulassung des PTV3-Konnektors ohne nonQES-XAdES-Signaturerstellung und Prüfung ermöglicht.	Neue Anforderung: A_18756 Optionalität von nonQES-XAdES-Signatur Der Konnektor KANN alle Aufrufe zu Signaturerstellung einer nonQES-XAdES-Signatur mit Fehler 4111 und alle Aufrufe zur Signaturprüfung einer nonQES-XAdES-Signatur mit Fehler 4112 beantworten. Die Signaturvarianten aus TAB_KON_778 werden damit weiter eingeschränkt. A_18756 wird dem Prüfverfahren funktionale Eignung "Herstellererklärung" zugewiesen.	gemSpec_Kon gemProdT_Kon_PTV3 gemProdT_Kon_PTV4
C_7133	gemSpec_Kon	Kapitel 4.1.2.1	**Es soll klargestellt werden, dass PDF/A-3 nicht unterstützt werden soll.**	Neue Anforderung: A_18780 - PDF/A-3 DARF NICHT unterstützt werden Der Konnektor DARF Dokumente im PDF/A-3 Format NICHT unterstützen. A_18780 wird dem Prüfverfahren funktionale Eignung "Test" zugewiesen.	gemSpec_Kon gemProdT_Kon_PTV3 gemProdT_Kon_PTV4

* Einige der Änderungen sind auch für den PTV4-Konnektor relevant. Die Anpassung des PTV4-Konnektors erfolgt im nächsten Wartungsrelease.

Entfernen der Prüfung von qualifizierten Zeitstempeln

Da auf absehbare Zeit in der TI keine Möglichkeit bestehen wird, qualifizierte Zeitstempel zu erstellen, wird eine Prüfung qualifizierter Zeitstempel durch den Konnektor aktuell nicht benötigt. Dabei wird auch die Inkonsistenz beseitigt, dass die Spezifikation nicht die Anforderung aus gemKPT_Arch_TIP umsetzt, da es sich um unterschiedliche Zeitstempel handelt.

Anpassungen in gemKPT_Arch_TIP:

TIP1-A_2255-01

[…]

~~In den *SignedData* enthaltene qualifizierte Zeitstempel werden ausgewertet.~~

Anpassungen in gemSpec_Kon:

4.1.8.1.3 Signaturzeitpunkt

Bezogen auf den vom Konnektor für die Signaturprüfung anzunehmenden Signaturerstellungszeitpunkt werden in dieser Spezifikation die Bezeichner Ermittelter_Signaturzeitpunkt und Benutzerdefinierter_Zeitpunkt verwendet.

Ermittelter_Signaturzeitpunkt: Vom Konnektor ermittelter Zeitpunkt, zu dem eine Signatur geprüft wird. Es werden folgende Signaturzeitpunkte ermittelt:

1. `Ermittelter_Signaturzeitpunkt_Eingebettet`:
   in der Signatur eingebetteter Zeitpunkt (falls vorhanden)
2. ~~`Ermittelter_Signaturzeitpunkt_Qualifiziert`:~~
   ~~qualifizierter Zeitstempel über die Signatur (falls vorhanden)~~
2. `Ermittelter_Signaturzeitpunkt_System`:
   Systemzeit des Konnektors bei Signaturprüfung

Anmerkung: Bei vom Konnektor selbst erstellten Signaturen ist immer ein in der Signatur eingebetteter Zeitpunkt vorhanden, jedoch kein qualifizierter Zeitstempel, da in der TI keine qualifizierten Zeitstempel ausgestellt werden. Sollte ein Dokument mit einem qualifizierten Zeitstempel versehen sein, so wird dieser nicht für die Ermittlung des Signaturzeitpunkts herangezogen.

Benutzerdefinierter_Zeitpunkt: Vom Benutzer beim Aufruf der Signaturprüfoperation als Parameter an den Konnektor übergebener Zeitpunkt, zu dem eine Signatur geprüft werden soll.

4.1.8.4.6 TUC_KON_151 „QES Dokumentensignatur prüfen“

TIP1-A_4672_01 - TUC_KON_151 „QES-Dokumentensignatur prüfen“

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

Tabelle 216: 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 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* (QES-signiertes Dokument vom Typ `QES_DocFormate` -> siehe Definition in Operation `VerifyDocument` mit `SIG:Document`) • signatureObject – *optional* (-> siehe Definition in Operation VerifyDocument mit dss:SignatureObject. Es werden Parallel- und Gegensignaturen unterstützt.) • optionalInputParams (optionale Eingabeparameter, siehe Operation VerifyDocument, Parameter SIG:OptionalInputs) • certificates – *optional/falls diese nicht im signierten Dokument enthalten sind, sondern nur referenziert werden* (X.509-Zertifikate). • xmlSchemas – *optional/nur für XML-Dokumente* (XMLSchema und ggf. weitere vom Hauptschema benutzte Schemata) • includeRevocationInfo [Boolean]: – *optional; Default: false* (Dieser optionale Parameter steuert die Einbettung von OCSP Antworten in die Signatur)
Komponenten	Konnektor
Ausgangsdaten	• verificationResult [VerificationResult] (Ergebnis der Signaturprüfung) • optionalOutput – *optional* (weitere Ausgabedaten gemäß SIG:OptionalOutput)
Standardablauf	**1. „DocumentValidation"**: Das signierte Dokument wird validiert mit Aufruf TUC_KON_080 „Dokument validieren"{ … }. Treten Fehler bei der Validierung der Typkonformität auf, wenn die Signatur im Dokument eingebettet ist, wird die Prüfung mit einem Fehler abgebrochen. 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.)

2. „CoreValidation“:

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.

XML-Signatur:
Die Core Validation erfolgt entsprechend [XMLDSig] Kapitel 3.2 Core Validation.
CMS-Signatur:
Die Core Validation erfolgt entsprechend Cryptographic Message Syntax (CMS) Kapitel 5.6 Signature Verification Process [RFC5652].
PDF-Signatur:
Die Core Validation erfolgt entsprechend [PAdES-3] Kapitel 4.6 Signature Validation aus PAdES-BES Part 3.

Auch wenn die Validierung fehlschlägt, werden die folgenden Prüfschritte durchgeführt, so dass ein vollständiges Prüfprotokoll erstellt werden kann.

3. „CheckSignatureCertificate“:

Teil 1: Signaturzertifikat ermitteln

XML-Signatur:
Das Signaturzertifikat ist im XMLDSig Element `ds:KeyInfo/ds:X509Data` gespeichert [XMLDSig] oder wird als Eingangsparameter übergeben.

CMS-Signatur:
Das Signaturzertifikat für CAdES ist im Feld `certificates` im `SignedData` Container gespeichert [CAdES] oder wird als Eingangsparameter übergeben.

PDF-Signatur:
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.

Teil 2: Signaturzeitpunkt bestimmen

Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_Eingebettet` wird wie folgt selektiert:

XML-Signatur:
Das XML element `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 7.2.1 XAdES [XAdES].

CMS-Signatur:
Das Attribut `SigningTime` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 11.3 CMS [CMS].

PDF-Signatur:
Der Signaturzeitpunkt kann dem M Eintrag des Signature Dictionary entnommen werden [PAdES Baseline Profile] Kapitel 6.2.1 Signing

time.
~~Der Signaturzeitpunkt~~
~~`Ermittelter_Signaturzeitpunkt_Qualifiziert` wird wie folgt selektiert:~~
~~XML-Signatur:~~
~~Das XML-element `SignatureTimeStamp` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 4.4.3.1 XAdES [XAdES].~~
~~CMS-Signatur und PDF-Signatur:~~
~~Das Attribut `signature-time-stamp` spezifiziert den Signaturzeitpunkt entsprechend Kapitel 6.1 CAdES [CAdES].~~
Der Signaturzeitpunkt `Benutzerdefinierter_Zeitpunkt` liegt gegebenenfalls als Aufrufparameter vor. Der Signaturzeitpunkt `Ermittelter_Signaturzeitpunkt_System` wird ermittelt.

Teil 3: Signaturzertifikatsprüfung:
Bei der folgenden Signaturzertifikatsprüfung sind die Signaturzeitpunkte gemäß [TIP1-A_5540] zu berücksichtigen.
Die Signaturzertifikatsprüfung erfolgt durch Aufruf von TUC_KON_037 „Zertifikat prüfen“ {
certificate = C.HP.QES;
qualifiedCheck = required;
baseTime = Signaturzeitpunkt;
offlineAllowNoCheck = true;
validationMode = OCSP;
ocspResponses = OCSP-Response;
getOCSPResponses = includeRevocationInfo
}.
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.
Sofern der Aufruf von TUC_KON_037 ocspResponses zurückgibt, wird die OCSP-Response des EE-Zertifikats in die Signatur eingebettet.
Auch wenn die Zertifikatsprüfung fehlschlägt, werden die folgenden Prüfungen durchgeführt.

4. „CheckPolicyConstraints“:

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“ und Tabelle TAB_KON_778 „Einsatzbereich der Signaturvarianten“ zu erfüllen.

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.

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. (->1) keine Signatur in signedDocument und signatureObject vorhanden: 4253. (→2 **„CoreValidation“**) Interner Fehler: 4001, Signatur des Dokuments ungültig: 4115, Signatur umfasst nicht das gesamte Dokument: 4262 (→3 **„CheckSignatureCertificate“**) Interner Fehler: 4001, Signaturzertifikat ermitteln ist fehlgeschlagen: 4206. (→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 218: TAB_KON_593 Übersicht Status für Prüfung einer Dokumentensignatur

VerificationResult für gesamtes Dokument (VerificationResult/HighLevelResult)	
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
VerificationResult pro Signatur (VerificationReport/IndividualReport/Result)	
Wert	Bedeutung mögliche Ausprägungen im VerificationReport
VALID	Die Signatur wurde gemäß den Regeln für die QES geprüft und für gültig befunden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:HasManifestResults
INVALID	Die Signatur ist ungültig oder aufgrund eines Fehlers konnte die Signaturprüfung nicht durchgeführt werden.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success

	ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	~~ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:Success~~ ~~ResultMinor =~~ ~~urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:InvalidSignatureTimestamp~~
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:invalid:IncorrectSignature
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:CertificateChainNotComplete
INCONCLUSIVE	Die Signatur wurde gemäß den Regeln für die QES geprüft. Allerdings konnten eine oder mehrere Prüfungen nicht vollständig durchgeführt werden. Einzelheiten finden sich in Result-Detail. Die Prüfungen, die durchgeführt werden konnten, waren erfolgreich.
	ResultMajor = urn:oasis:names:tc:dss:1.0:resultmajor:InsufficientInformation ResultMinor = urn:oasis:names:tc:dss:1.0:resultminor:OcspNotAvailiable Hinweis: Das Erreichen dieses Zustandes hängt davon ab, ob eine OCSP-Abfrage nicht durchgeführt werden konnte, unabhängig davon, ob die Ursache dafür die Offlineschaltung des Konnektors (MGM_LU_ONLINE = Disabled) oder die Nichterreichbarkeit des OCSP-Responders im Online-Betrieb (MGM_LU_ONLINE = Enabled) ist.

[<=]

TIP1-A_5540_01 - QES-Signaturprüfergebnis bezogen auf Signaturzeitpunkt
Der Konnektor MUSS zur QES-Signaturprüfung ein Prüfergebnis, das sich auf genau einen angenommenen Signaturzeitpunkt bezieht, an den Aufrufer zurückgeben.
Die Auswahl des angenommenen Signaturzeitpunkts, auf den sich das Signaturergebnis bezieht, erfolgt hierarchisch:

- `Benutzerdefinierter_Zeitpunkt`
  falls vorhanden, sonst
- `Ermittelter_Signaturzeitpunkt_Eingebettet`
  falls vorhanden, sonst
- ~~`Ermittelter_Signaturzeitpunkt_Qualifiziert`~~
  ~~falls vorhanden, sonst~~
- `Ermittelter_Signaturzeitpunkt_System`

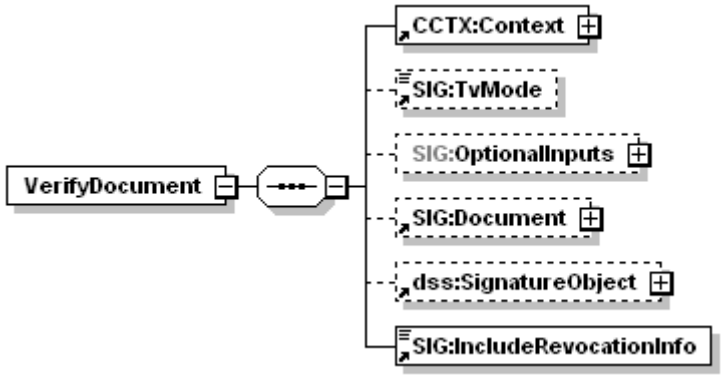
[<=]

4.1.8.5.2 VerifyDocument (nonQES und QES)

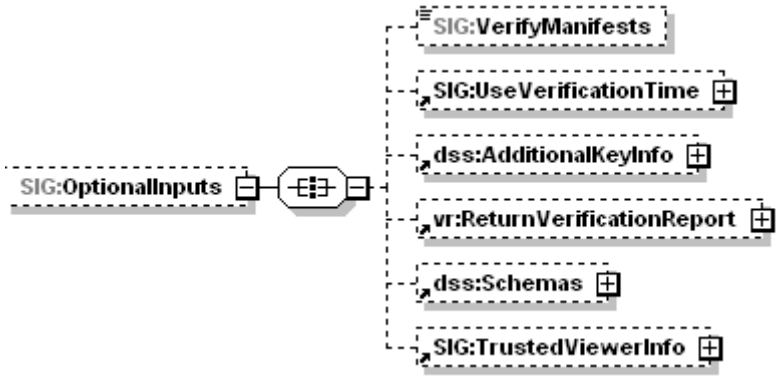
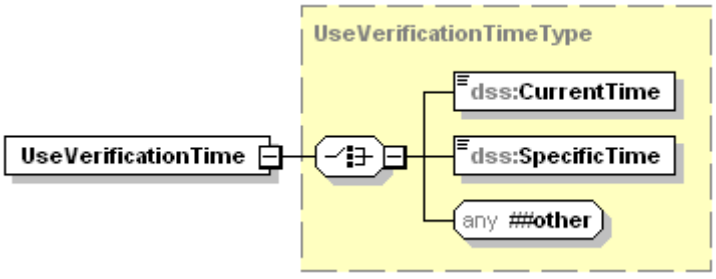
TIP1-A_5034-03 - Operation VerifyDocument (nonQES und QES)

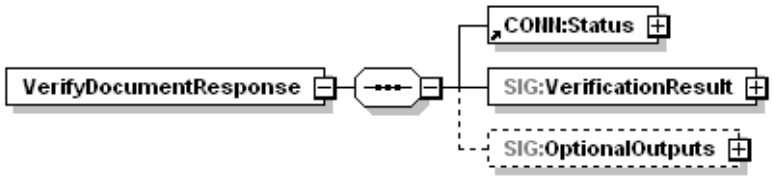
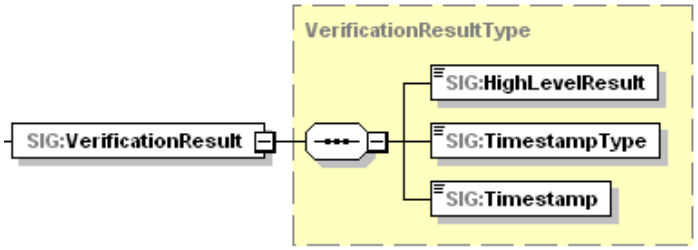
Der Signaturdienst des Konnektors MUSS an der Clientschnittstelle eine an [OASIS-DSS] angelehnte Operation VerifyDocument (nonQES und QES) anbieten.

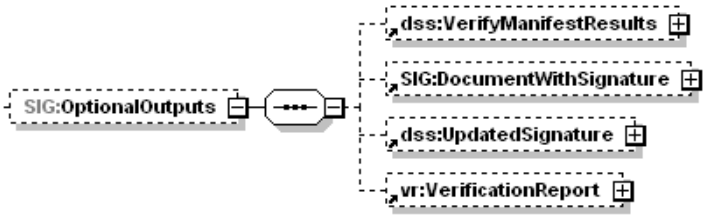
Tabelle 223: TAB_KON_066 Operation VerifyDocument (nonQES und QES)

<table>
<tr><td>Name</td><td colspan="2">VerifyDocument</td></tr>
<tr><td>Beschreibung</td><td colspan="2">Diese Operation verifiziert die Signatur eines Dokumentes.
Der Konnektor MUSS jede konform zur Außenschnittstelle SignDocument erzeugte Signatur durch VerifyDocument prüfen können. ~~Darüber hinaus müssen im Fall QES, falls vorhanden, auch qualifizierte Zeitstempel geprüft werden.~~ Außerdem MÜSSEN die zusätzlich geforderten Signaturverfahren zur Dokumentensignaturprüfung unterstützt werden
Das Ergebnis der Prüfung wird, wenn gefordert, in Form eines standardisierten Prüfberichts in einer VerificationReport-Struktur gemäß [OASIS-VR] zurückgeliefert.</td></tr>
<tr><td>Aufruf-parameter</td><td colspan="2"> 

<table>
<tr><td>Name</td><td>Beschreibung</td></tr>
<tr><td>CCTX:
Context</td><td>MandantId, ClientSystemId, WorkplaceId verpflichtend;
UserId nicht ausgewertet</td></tr>
<tr><td>TvMode</td><td>Der Parameter wird im Konnektor nicht ausgewertet.</td></tr>
<tr><td>SIG:
Optional
Inputs</td><td>Enthält optionale Eingabeparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7):
Die zulässigen optionalen Eingabeparameter sind unten erläutert.</td></tr>
<tr><td>SIG:
Document</td><td>Enthält im Fall der Prüfung von detached oder enveloped Signaturen das zur Signatur gehörende bzw. das diese umschließende Dokument (siehe [OASIS-DSS] Section 2.4.2 und oben).</td></tr>
<tr><td>dss:
Signa
ture
Object</td><td>Enthält die zu prüfende Signatur, wenn sie nicht im Dokument selbst eingebettet ist ([OASIS-DSS] Kapitel 4.1).
Hierbei werden XML-Signaturen als ds:Signature Element und alle anderen Signaturen als dss:Base64Signature mit entsprechend gesetztem Type-Attribut (siehe SignatureType, Operationen SignDocument und ExternalAuthenticate) übergeben, wobei die nachfolgenden Werte unterstützt werden müssen:
• CMS-Signatur</td></tr>
</table></td></tr>
</table>

		urn:ietf:rfc:5652 • **S/MIME-Signatur** urn:ietf:rfc:5751 • **PDF-Signatur** http://uri.etsi.org/02778/3 • **PKCS#1-Signatur** (siehe Operation ExternalAuthenticate) urn:ietf:rfc:3447 • **ECC-Signatur** (siehe Operation ExternalAuthenticate) urn:bsi:tr:03111:ecdsa
	`SIG: IncludeRevocationInfo`	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturprüfung vorliegenden Sperrinformationen anfordern. Ist bereits eine Sperrinformation eingebettet, so wird die neue Sperrinformation zusätzlich eingebettet. Für in einer Gegensignatur enthaltene Signaturen erfolgt keine Einbettung von Sperrinformationen. Für PDF-Signaturen erfolgt keine Einbettung von Sperrinformationen. Der Konnektor nimmt die Warnung 4261 in die Antwort auf.
		
	`SIG: VerifyManifests`	Durch das in [OASIS-DSS] (Abschnitt 4.5.1) definierte Element kann die Prüfung eines ggf. vorhandenen Manifests angefordert werden.
		
	`SIG: UseVerification`	Durch das in [OASIS-DSS] (Abschnitt 4.5.2) spezifizierte Element kann die Prüfung der Signatur bezüglich eines durch den Aufrufer bestimmten Zeitpunktes (`Benutzerdefinierter_Zeitpunkt`) erfolgen.

	Time	
	dss:AdditionalKeyInfo	Durch das in [OASIS-DSS] (Abschnitt 4.5.4) spezifizierte Element kann zusätzliches, für die Prüfung benötigtes, Schlüsselmaterial übergeben werden.
	vr:ReturnVerificationReport	Durch dieses in [OASIS-VR] spezifizierte Element kann die Erstellung eines ausführlichen Prüfberichtes angefordert werden. Der Konnektor MUSS die Anforderungen der Konformitätsstufe 2 („Comprehensive“) erfüllen und die Profilierung aus Anhang B3 beachten.
	dss:Schemas	Durch das in [OASIS-DSS] (Abschnitt 2.8.5) definierte Element können eine Menge von XML-Schematas übergeben werden, die zur Validierung des übergebenen XML-Dokumentes verwendet werden können. Zur Struktur dieses Elements siehe Beschreibung des Parameters `dss:Schemas` der Operation SignDocument.
	SIG:ViewerInfo	Der Parameter wird im Konnektor nicht ausgewertet.
Rückgabe		
	Status	Enthält den Ausführungsstatus der Operation.
	SIG:VerificationResult	 Das Element `Sig:VerificationResult` enthält das Ergebnis der Prüfung als Ampel, den Typ des zugehörigen angenommenen Signaturzeitpunkts und der angenommene Signaturzeitpunkt selbst.
	SIG:HighLevelResult	Das Ergebnis der Prüfung (Ampelschaltung) mit folgenden Werten: • VALID: alle Signaturen sind gültig • INVALID: mindestens eine der Signaturen ist ungültig • INCONCLUSIVE: in allen anderen Fällen
	SIG:TimestampType	Der Typ des angenommenen Signaturzeitpunkts mit folgenden Werten: • SIGNATURE_EMBEDDED_TIMESTAMP: in der Signatur eingebetter Zeitpunkt

		`Ermittelter_Signaturzeitpunkt_Eingebettet` ~~• QUALIFIED_TIMESTAMP: qualifizierter Zeitstempel über die Signatur `Ermittelter_Signaturzeitpunkt_Qualifiziert`~~ • SYSTEM_TIMESTAMP: Systemzeit des Konnektors bei Signaturprüfung `Ermittelter_Signaturzeitpunkt_System` • USER_DEFINED_TIMESTAMP: benutzerdefinierter Zeitpunkt `Benutzerdefinierter_Zeitpunkt` Als Format darf jedes zum XML-Typ "dateTime" konforme Format verwendet werden (<element name="Timestamp" type="dateTime"/>). Wenn mehrere Signaturen im Dokument vorhanden sind, wird hier der angenommene Signaturzeitpunkt der jüngsten Signatur angegeben.
	`SIG: Timestamp`	Im Element SIG:Timestamp wird der zu SIG:TimestampType gehörende Zeitstempel zurückgegeben.
	`SIG: OptionalOutputs`	Enthält (angelehnt an dss:OptionalOutputs, wie in Abschnitt 2.7 von [OASIS-DSS] beschrieben) optionale Ausgangselemente: 
	`dss: VerifyManifestResults`	Dieses in Abschnitt 4.5.1 von [OASIS-DSS] definierte Element enthält Informationen zur Prüfung eines ggf. vorhandenen Signaturmanifests und wird zurückgeliefert, sofern beim Aufruf das `dss:VerifyManifest`-Element, aber nicht das `RequestVerificationReport` als optionales Eingabeelement übergeben wurde.
	`SIG: DocumentWithSignature`	Dieses in Abschnitt 4.5.8 von [OASIS-DSS] spezifizierte Element wird zurückgeliefert, falls eine in dem Dokument enthaltene Signatur (Enveloped Signature) in Verbindung mit dem `SIG:IncludeRevocationInfo`-Element geprüft wurde.
	`dss: UpdatedSignature`	Dieses in Abschnitt 4.5.8 von [OASIS-DSS] spezifizierte Element wird zurückgeliefert, falls eine abgesetzte (Detached Signature) oder umschließende (Enveloping Signature) in Verbindung mit dem `SIG:IncludeRevocationInfo`-Element geprüft wurde.
	`vr: Verification`	Dieses in [OASIS-VR] spezifizierte Element wird zurückgeliefert, falls das `ReturnVerificationReport`-Element als Eingabeparameter verwendet wurde. Die

	`Report`	Profilierung von Anhang B3 MUSS beachtet werden.
Vorbe-dingungen	Keine	
Nachbe-dingungen	Keine	

Tabelle 224: TAB_KON_760 Ablauf Operation VerifyDocument (nonQES und QES)

Nr.	Aufruf Technischer Use Case oder Interne Operation	Beschreibung
1.	checkArguments	Die übergebenen Werte werden auf Konsistenz und Gültigkeit überprüft. Treten hierbei Fehler auf, so bricht die Operation mit Fehler 4000 ab.

2.	TUC_KON_000 „Prüfe Zugriffs- berechtigung“	Die Prüfung erfolgt durch den Aufruf
		TUC_KON_000 {
		mandantId = $context.mandantId;
		clientsystemId = $context.clientsystemId;
		workplaceId = $context.workplaceId;
		needCardSession= false;
		}
		Tritt bei der Prüfung ein Fehler auf, bricht die Operation mit Fehlercode aus TUC_KON_000 ab.

3.	prüfe, ob QES oder nonQES	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-Signatur, andernfalls liegt eine nonQES-Signatur vor.
Für QES-Signaturen wird Schritt 4 ausgeführt. Für nonQES-Signaturen wird Schritt 5 ausgeführt.		
4.a	Prüfe Signaturdienst-Modul	Prüfe, ob MGM_LU_SAK=Enabled. Ist dies nicht der Fall, so bricht die Operation mit Fehler 4125 ab.
4.b	TUC_KON_151 „QES Dokumentensignatur prüfen“	Die QES wird geprüft. Tritt hierbei ein Fehler auf, bricht die Operation ab.
5.	TUC_KON_161 „nonQES Dokumentensignatur prüfen“	Die nonQES wird geprüft. Tritt hierbei ein Fehler auf, bricht die Operation ab.

6.3 Profilierung VerificationReport

…

Sonderfälle:

Dokument mit parallelen Signaturen

Für jede Signatur wird ein IndividualReport erzeugt.

Dokument mit Signatur und Gegensignatur

Für jede Signatur wird ein IndividualReport erzeugt.

~~**Dokument mit Signatur und qualifiziertem Zeitstempel**~~

~~- Für den Zeitstempel wird ein eigener IndividualReport mit IndividualTimeStampReport erzeugt.~~