
C_12408_Anlage

Inhaltsverzeichnis

1 Änderungsbeschreibung.....	2
2 Änderung in gemSpec_IDP_Sek.....	3
2.1 Kapitel "4.3 Identifizierung und Authentifizierung des Nutzers" wird angepasst.....	3
2.2 Kapitel "4.3.2 Authentifizierungsverfahren" wird angepasst.....	3
2.3 Kapitel "4.2.4.2 Token-Endpunkt Ausgangsdaten" wird angepasst.....	5
3 Änderungen in Steckbriefen.....	37
3.1 Änderungen in gemProdT_IDP-Sek_PTV.....	37
3.2 Änderungen in gemAnbT_IDP-Sek_KTR_ATV.....	37

1 Änderungsbeschreibung

Es besteht der Bedarf für ein Authentisierungsverfahren „eGK ohne PIN“ für die GesundheitsID auf einem Vertrauensniveau „gematik-ehealth-loa-normal“. Dazu muss Folgendes gewährleistet sein:

- Die Definition des Vertrauensniveaus „gematik-ehealth-loa-normal“ muss erfolgen.
- Sektorale IDPs müssen die Möglichkeit haben, „eGK ohne PIN“ als Authentisierungsmittel zuzulassen und zu implementieren.

2 Änderung in gemSpec_IDP_Sek

2.1 Kapitel "4.3 Identifizierung und Authentifizierung des Nutzers" wird angepasst

Neu - Definition des Vertrauensniveau "gematik-ehealth-loa-normal"

A_28137 -Definition "gematik-ehealth-loa-normal"

Der Anbieter des sektoralen IDP MUSS gematik-ehealth-loa-normal wie folgt interpretieren:

Der Wert gematik-ehealth-loa-normal entspricht der Widerstandsfähigkeit des Authentisierungsmittels und Protokolls gegen das Angriffspotential "Enhanced-Basic" nach [ISO18045]. Zertifizierungen, Notifizierungen oder Bestätigungen von Prozessen oder Prozessbestandteilen vergleichbarer Normen und Richtlinien, z. B. nach Verordnung (EU) Nr. 910/2014 in Verbindung mit (EU) 2015/1502 an elektronische Identifizierungsmittel, BSI TR-03107-1 oder vergleichbar, können nachgenutzt werden. [\leq , Anb_IDP-Sek_KTR, Sich.techn. Eignung: Gutachten (Anbieter)]

2.2 Kapitel "4.3.2 Authentifizierungsverfahren" wird angepasst

Alt:

A_23129-05 -Identifikation des Authentisierungsverfahren

Der sektorale IDP MUSS den Claimamr im ID Token entsprechend dem durchgeführten Authentisierungsverfahren nach folgender Tabelle befüllen.

Tabelle 1: Codierung der Authentisierungsverfahren

Authentisierungsverfahren	Wert desamr Claim	zulässiges Niveau (acr)
Authentisierung mittels eGK und PIN	urn:telematik:auth:eGK	gematik-ehealth-loa-high
Authentisierung mittels elektronischen Identitätsnachweises(Online-Ausweisfunktion)	urn:telematik:auth:eID	gematik-ehealth-loa-high
Authentisierung mittels eGK und PIN ohne Prüfung Gerätebindung (Gastzugang)	urn:telematik:auth:guest:eGK	gematik-ehealth-loa-high
Anderes Authentisierungsverfahren	urn:telematik:auth:other	gematik-ehealth-loa-high und

		gematik-ehealth-loa-substantial
--	--	---------------------------------

【<=, IDP-Sek, Sich.techn. Eignung: Produktgutachten】

Neu:

A_23129-06 -Identifikation des Authentisierungsverfahren

Der sektorale IDP MUSS den Claimamr im ID_TOKENentsprechend dem durchgeführten Authentisierungsverfahren nach folgender Tabelle befüllen.

Tabelle 2: Codierung der Authentisierungsverfahren

Authentisierungsverfahren	Wert desamr Claim	zulässiges Niveau (acr)
Authentisierung mittels eGK und PIN	urn:telematik:auth:eGK	gematik-ehealth-loa-high
Authentisierung mittels elektronischem Identitätsnachweis (Online-Ausweisfunktion)	urn:telematik:auth:eID	gematik-ehealth-loa-high
Authentisierung mittels eGK und PIN ohne Prüfung Gerätebindung (Gastzugang)	urn:telematik:auth:guest:eGK	gematik-ehealth-loa-high
Authentisierung mittels eGK ohne PIN	urn:telematik:auth:touch:eGK	gematik-ehealth-loa-normal
Authentisierung mittels eGK ohne PIN ohne Prüfung einer GesundheitsID	urn:telematik:auth:guest:touch:eGK	gematik-ehealth-loa-normal
Anderes Authentisierungsverfahren	urn:telematik:auth:other	gematik-ehealth-loa-high und gematik-ehealth-loa-substantial

【<=, IDP-Sek, Sich.techn. Eignung: Produktgutachten】

2.3 Kapitel "4.2.4.2 Token-Endpunkt Ausgangsdaten" wird angepasst

Neue Anforderung ID_TOKEN bei eGK ohne PIN

A_28229 -Inhalte des ID_TOKEN nach Authentifizierung mit eGK ohne PIN

Der Hersteller eines sektoralen IDP MUSS sicherstellen, dass im ausgestellten ID_TOKEN nach einer Authentifizierung einer eGK ohne PIN ohne angelegte GesundheitsID folgende Claims belegt sind:

Claim	Wert
acr	"gematik-ehealth-loa-normal"
amr	"urn:telematik:auth:guest:touch:eGK"
urn:telematik:claims:id	KVNR des Versicherten, ausgelesen aus der eGK
urn:telematik:claims:organization	IK-Nummer der Krankenkasse, ausgelesen aus der eGK

【<=, IDP-Sek, Sich.techn. Eignung: Produktgutachten】

Neues Kapitel 5.4 eGK-Handling:

Neues Kapitel 5.4.1 eGK-Handling, Einführung (aus gemSpec_PoPP-Service)

In diesem Kapitel wird beschrieben und spezifiziert, wie der sektorale IDP Nachrichten mit einer Smartcard austauscht. Der Transport solcher Nachrichten wird in anderen Kapiteln behandelt. Deshalb ist das Kommunikationsmodell hier einfach: Der sektorale IDP schickt Kommando-APDU zu einer Smartcard und erhält von dort die korrespondierenden Antwort-APDU zur Auswertung.

Der IDP kommuniziert im Rahmen des folgenden Use Cases mit einer Smartcard:

- Ein Versicherter nutzt beim mobilen Check-in ein Versichertenendgerät. Das Versichertenendgerät verfügt über einen Standard-Kartenleser mit dem die eGK:
 - a. kontaktbehaftet kommuniziert oder
 - b. kontaktlos kommuniziert.

Im Rahmen der Authentifizierung einer eGK ohne Eingabe einer PIN verfolgt der sektorale IDP bei der Kommunikation mit einer Smartcard die folgenden Ziele:

1. Der sektorale IDP überzeugt sich, dass es sich bei der Smartcard um eine echte eGK handelt.
2. Der sektorale IDP überzeugt sich, dass die eGK gültig ist.
3. Der sektorale IDP liest aus der eGK Daten aus.
 - a. Wenn für die Versicherten keine GesundheitsID im sektoralen IDP angelegt ist übernimmt dieser die ausgelesenen Daten in das ID_TOKEN. In diesem Fall dient die eGK als Authorisierungsmittel und Datenquelle.
 - b. Wenn für die Versicherten GesundheitsID im sektoralen IDP angelegt ist übernimmt dieser die Daten der GesundheitsID des Versicherten in das ID_TOKEN. In diesem Fall dient die eGK ausschließlich als Authorisierungsmittel.

Die genannten Ziele werden bei kontaktloser Kartenkommunikation mit einer eGK basierend auf [gemSpec_eGK_ObjSys_G2.1](die einen PACE Kanal voraussetzt) wie folgt erreicht:

1. Der sektorale IDP authentisiert die eGK mit der Identität ID.C.eGK.AUT_CVC.E256. Wegen [gemSpec_COS#N107.235)b] ist es nicht möglich dabei einen Trusted Channel zwischen dem sektoralen IDP und der eGK zu etablieren.
2. Der sektorale IDP liest aus der eGK das X.509-Zertifikat aus der Datei EF.C.CH.AUT.E256 aus und überprüft dieses auf Gültigkeit. Da der Kommunikationskanal zwischen sektoralen IDP und eGK im kontaktlosen Fall nicht Ende-zu-Ende gesichert ist, ist der sektorale IDP nicht ohne weiteres in der Lage zu beurteilen, ob das präsentierte X.509-Zertifikat von derselben eGK stammt, deren Echtheit er mit der Identität ID.C.eGK.AUT_CVC.E256 überprüft hat. Deshalb konsultiert der sektorale IDP im kontaktlosen Fall eine Datenbank, welche die Frage beantwortet: Stammen das CV-Zertifikat der Echtheitsprüfung und das präsentierte X.509-Zertifikat aus ein und derselben eGK? So eine Datenbank wird im Kapitel "eGK-Hash Datenbank" beschrieben.
3. Der sektorale IDP entnimmt dem präsentierten X.509-Zertifikat die Daten je nach Konstellation für das Auffinden der GesundheitsID oder für die Befüllung des ID_TOKEN.

Der sektorale IDP schaltet in der eGK nichts frei und erwartet auch nicht, dass in der eGK etwas freigeschaltet ist, insbesondere weder durch Card-2-Card noch durch eine PIN-Eingabe. Technisch ist dies gleichbedeutend mit der Aussage, dass der sektorale IDP nur

solche Kommando-APDU an eine eGK sendet, für die im Rahmen der Objektsystemspezifikation die Zugriffsbedingung "ALWAYS" festgelegt ist ("ALWAYS" = jeder, der im Besitz der Karte ist, ist in der Lage diese Operation auszuführen).

Hinweis: Im kontaktlosen Fall funktioniert eine sinnvolle Kartenkommunikation mit der eGK nur nach Aufbau eines PACE-Kanals. Weil die dazu notwendige CAN auf der eGK aufgedruckt ist und somit jeder, der im Besitz der eGK ist so einen PACE-Kanal aufzubauen in der Lage ist, wird hier der Einfachheit halber die Etablierung eines PACE-Kanals und die damit verbundene Freischaltung von Funktionalität in der eGK auch unter "ALWAYS" subsumiert.

Hinweis: Im kontaktlosen Fall stehen nach Etablierung eines PACE-Kanals dieselben Daten und Funktionen in einer eGK zur Verfügung, wie im kontaktbehafteten Fall unmittelbar nach Stecken einer eGK.

Hinweis: Für die Generation 3 einer eGK ist geplant, dass eine eGK G3 eine Identität besitzt, die sich ohne PIN-Eingabe nutzen lässt und deren zugehöriges X.509-Zertifikat alle Informationen enthält, die für ein für die Ermittlung der GesundheitsID im sektoralen IDP relevant sind.

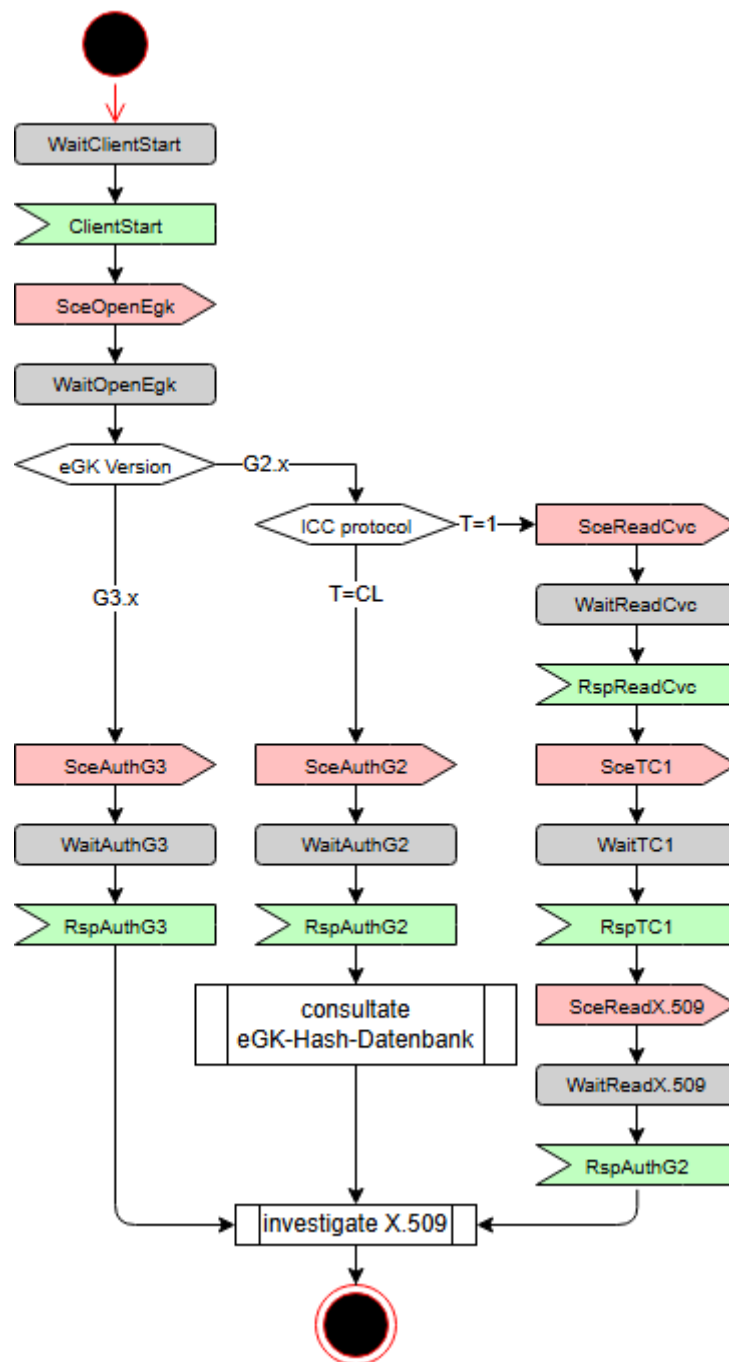


Abbildung 1 : Zustandsdiagramm für die Verarbeitung einer eGK

Abbildung "Zustandsdiagramm für die Verarbeitung einer eGK" zeigt das Zustandsdiagramm für die Verarbeitung einer eGK. Der sektorale IDP wartet in jedem der gezeigten Zustände auf den Empfang einer Nachricht. Er bearbeitet die Nachricht, sendet eine passende Nachricht zurück und geht in den Folgezustand über. Berücksichtigt sind die kontaktbehaftete und die kontaktlose Handhabung einer eGK Generation (G)2.x, sowie die (geplante) Handhabung einer eGK G3 (für welche die Handhabung kontaktbehaftet und kontaktlos identisch ist). Für den sektoralen IDP und die Nutzung der eGK ohne PIN als Authentisierungsmittel ist in erster Linie der kontaktlose Ablauf relevant. Das Zustandsdiagramm berücksichtigt nur Gutfälle. Das bedeutet, Fehlerfälle, Abbrüche und ähnliches sind im abgebildeten Zustandsdiagramm nicht enthalten.

Neues Kapitel 5.4.2 Szenario

In diesem Kapitel ist "Szenario" definiert als eine Abfolge von Elementen in einer Liste. Jedes Element enthält eine Kommando-APDU und eine Liste von Statuswörtern, die als Gutfall für eine Antwort-APDU gewertet werden.

Definition CommandApdu: Eine Kommando-APDU gemäß [gemSpec_COS].

Definition ExpectedStatusWord: Eine Menge mit einem oder mehreren Statuswörtern aus [gemSpec_COS].

Definition Step: Eine Aggregation von genau einer CommandApdu und einem Objekt ExpectedStatusWord.

Definition Szenario: Eine Liste mit keinem, einem oder mehreren Elementen des Typs ScenarioStep.

analog PoPP A_27000

A_28383 -Sektoraler IDP: StandardScenarioMessage

Der sektorale IDP MUSS Objekte vom Typ StandardScenarioMessage generieren, wie in der Schnittstellspezifikation [todo.yaml] beschrieben, wobei das Attribut "steps" ein Szenario ist. [≤, ,]

analog PoPP A_27017

A_28384 -Sektoraler IDP: Erlaubnis für abweichende Szenarien

Falls der sektorale IDP Szenarien verwendet, die von den in [gemSpec_IDP_Sek] beschriebenen abweichen, so MUSS der Hersteller des sektoralen IDP vor deren Verwendung eine Erlaubnis der gematik einholen. [≤, ,]

analog PoPP A_27129

A_28385 -Sektoraler IDP: Codierung von Nicht-Konnektor-Szenarien

Der sektorale IDP MUSS für die Codierung das Szenario in eine StandardScenarioMessage einstellen. [≤, ,]

Hinweis zu A_28385: Die Beschreibung des StandardScenarioMessage folgt noch in einer OpenAPI-Spezifikation!

Neues Kapitel 5.4.2.1 eGK öffnen

Dieses Kapitel beschreibt, wie ermittelt wird, ob es sich bei der präsentierten Smartcard um eine eGK handelt und welche Version diese hat.

Die gematik stellt mit der Prüfkarte eGK eine elektronische Identität zur Überprüfung verschiedener Anwendungsfälle in der TI zur Verfügung. Sie wird vorrangig von Dienstleistern vor Ort (DVOs) genutzt. Die Prüfkarte eGK ist nicht für die Nutzung im regulären Versorgungsalltag von Leistungserbringern oder Versicherten vorgesehen. Um sie nutzen zu können, muss sie vom sektoralen IDP wie eine normale eGK behandelt werden.

Definition: eGKIncludedPtvObjSys ist eine Menge mit Produkttypversionen von eGK-Objektsystemen, die der sektorale IDP zu unterstützen hat. Produkttypversionen werden in diese Menge aufgenommen oder aus ihr entfernt, wenn sich die Anforderungslage (also die Vorgaben der gematik) ändert. Basierend auf den Erfahrungen der Vergangenheit ist davon auszugehen, dass sich diese Menge nur wenige Male pro Jahr ändert, während sich die Anzahl zugelassener Kartenprodukte häufiger wechselt. Deshalb wird die Menge zulässiger Kartenprodukte über die Produkttypversion definiert.

Definition: eGKexcludedPiObjSys ist eine Menge mit Produktidentifikationen aktiver Objektsysteme, die von der Verwendung durch den sektoralen IDP ausgeschlossen werden. Produktidentifikationen der Kartenhersteller werden in diese Menge aufgenommen, wenn es Schwierigkeiten mit einem konkreten Kartenprodukt gibt. Basierend auf den Erfahrungen der Vergangenheit ist davon auszugehen, dass die Mächtigkeit der Menge klein ist und sich nur selten ändert.

analog PoPP A_28040

A_28386 -Sektoraler IDP: Verarbeiten von Prüfkarte eGK

Der sektorale IDP MUSS eine eGK-Prüfkarte wie eine normale eGK behandeln. [<=, ,]

analog PoPP A_27008

A_28387 -Sektoraler IDP: Szenario SceOpenEgk, eGK öffnen

Der PoPP-Service MUSS im ersten Szenario folgende Liste verwenden:

1. Element:

- a. CommandApdu: '00 a4 040c 07 D27600001448000'
- b. ExpectedStatusWord: {'9000'}

2. Element:

- a. CommandApdu: '00 b0 9100 00'
- b. ExpectedStatusWord: {'9000', '6281'}

Hinweis 1: Das erste Listenelement selektiert das Master File (MF) einer eGK. Dieses Kommando bringt eine eGK in einen für die nachfolgenden Kommandos definierten Zustand. Antwortet die Smartcard auf dieses Kommando mit einem erwarteten Statuswort, dann handelt es sich um eine eGK (oder um eine Smartcard, die vorgaukelt eine eGK zu sein). Wird irrtümlich eine Karte mit falschem Typ angesprochen (etwa ein HBA oder eine Bankkarte), dann wird das durch ein inakzeptables Statuswort erkannt. Falls eine Karte vorgaukelt eine eGK zu sein, dann wird dies in einem späteren Szenario erkannt.

Hinweis 2: Das zweite Listenelement liest den Inhalt von EF.Version2. Basierend auf den Versionsinformationen ist es möglich eGK unterschiedlicher Generationen zu erkennen, oder beispielsweise wegen Schwachstellen abzulehnen. [<=, ,]

analog PoPP A_27018

A_28388 -Sektoraler IDP: Zulässige eGK-Objektsystemversionen

Der sektorale IDP MUSS Änderungen an der Menge eGKincludedPtvObjSys, die ihm ausschließlich durch die gematik angezeigt werden, innerhalb von sieben Tagen im Betrieb berücksichtigen. [<=, ,]

analog PoPP A_27019

A_28389 -Sektoraler IDP: Unzulässige eGK-Objektsystemversionen

Der sektorale IDP MUSS Änderungen an der Menge eGKexcludedPiObjSys, die ihm ausschließlich durch die gematik angezeigt werden, innerhalb von 24 Stunden im Betrieb berücksichtigen.

Hinweis 1: Die Produkttypversion eines Kartenproduktes findet sich in der Datei EF.Version2. Basierend auf der Anforderungslage der gematik besteht die Menge eGKincludedObjSys im März 2025 aus folgenden Elementen:

{'040400', '040401', '040500', '040501', '040502', '040600', '040700'}.

Mit Einführung der eGK G3 wird die Menge (laut aktuellem Plan) um den Wert '050000' ergänzt.

Hinweis 2: Die Produktidentifikation des aktiven Objektsystems findet sich in der Datei EF.Version2. Basierend auf dem Kenntnisstand März 2025 ist die Menge eGKexcludedObjSys leer. [<=, ,]

angepasste Afo nach PoPP A_27009

A_28390 -Sektoraler IDP: Auswertung SceOpenEgk

Der sektorale IDP MUSS die Kartenantworten auf das Szenario SceOpenEgk wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceOpenEgk beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Der Inhalt der Datei EF.Version2 wird gemäß [gemSpec_Karten_Fach_TIP_G2.1#2.1.1] ausgewertet:
 - a. Falls die Version des aktiven Objektsystems (PT_ObjSys) nicht Element der Menge eGKIncludedPtvObjSysist, dann bricht der Use Case mit der Fehlermeldung InvalidPtvObjectSystem ab.
 - b. Falls die Produktidentifikation des aktiven Objektsystems (PI_Objektsystem) Element der Menge eGKexcludedPiObjSys ist, dann bricht der Use Case mit der Fehlermeldung InvalidPiObjectSystem ab.
 - c. Falls die Version des aktiven Objektsystems (PT_ObjSys) eine eGK der Generation 2 anzeigt, wird mit dem Szenario SceAuthG2 aus [A_28391*] fortgefahren.
 - d. Zeigt sie eine Generation 3 an, wird mit dem Szenario SceAuthG3 aus [A_28393*] fortgefahren.

Hinweis: SceAuthG2 gilt nur für den kontaktlosen Fall (NFC). Das Szenario kontaktbehaftete eGK wird für den mobilen Einsatz derzeit nicht betrachtet. [<=, ,]

Neues Kapitel 5.4.2.2 eGK G2 kontaktlos

Dieses Kapitel behandelt die kontaktlose Kommunikation einer eGK gemäß [gemSpec_eGK_ObjSys_G2.1]. Dies deckt den Anwendungsfall "Versicherter mit Versichertenendgerät mit NFC-Schnittstelle zum Auslesen der eGK" ab.

Im Gutfall schickt der sektorale IDP nach dem in [A_28387*] beschriebenen Szenario ein weiteres Szenario mit einer Reihe von Kommando-APDU an die Karte:

analog PoPP A_27020

A_28391 -Sektoraler IDP: Szenario SceAuthG2

Der sektorale IDP MUSS bei kontaktloser Kommunikation mit einer eGK G2 im zweiten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 b0 8700 00'
 - b. ExpectedStatusWord: {'9000', '6281'}
2. Element:
 - a. CommandApdu: '00 b0 8600 00'
 - b. ExpectedStatusWord: {'9000', '6281'}
3. Element:
 - a. CommandApdu: '00 a4 040c 0a a0000000167455349474e'
 - b. ExpectedStatusWord: {'9000'}
4. Element:
 - a. CommandApdu: '00 22 41A4 06 (84-01-09) || (80-01-00)'

- b. ExpectedStatusWord: {'9000'}
- 5. Element:
 - a. CommandApdu: '00 b0 8400 00 0000'
 - b. ExpectedStatusWord: {'9000', '6281'}
- 6. Element:
 - a. CommandApdu: '00 88 0000 18 token 00'
 - b. ExpectedStatusWord: {'9000'}.

Hinweis 1: Das erste Listenelement liest den Inhalt von EF.C.CA.CS.E256 mit CVC-Sub-CA aus. Der sektorale IDP benötigt dieses CV-Zertifikat zur Verifikation des End-Entity-CVC (sofern er es nicht aus anderen Quellen bereits kennt).

Hinweis 2: Das zweite Listenelement liest den Inhalt von EF.C.eGK.AUT_CVC.E256 mit dem End-Entity-CVC aus. Der sektorale IDP benötigt dieses CV-Zertifikat für die Authentisierung.

Hinweis 3: Das dritte Listenelement selektiert das Verzeichnis DF.ESIGN.

Hinweis 4: Das vierte Listenelement ist ein MSE Set Kommando gemäß

[gemSpec_COS#(N100.900)] zur Selektion des privaten Schlüssels

PrK.eGK.AUT_CVC.E256 für den Algorithmus elcRoleAuthentication.

Hinweis 5: Das fünfte Listenelement liest das X.509-Zertifikat aus der Datei EF.CH.AUT.E256.

Hinweis 6: Das sechste Listenelement ist ein INTERNAL AUTHENTICATE Kommando gemäß [gemSpec_COS#(N086.400)].[<=, ,]

analog PoPP A_27021

A_28392 -Sektoraler IDP: Auswertung SceAuthG2

Der sektorale IDP MUSS die Kartenantworten auf das Szenario SceAuthG2 wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceAuthG2 beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Der Use Case wird mit der Fehlermeldung InvalidCaCvc beendet, wenn eine Prüfung des CV-Zertifikats gemäß [gemSpec_COS#(N095.900)b] aus der Datei EF.C.CA.CS.E256 gegen das zugehörige CVC-Root-CA aus der TSL-Liste mit einem Fehler endet, dabei gilt:
 - a. affectedObjectaus [gemSpec_COS#(N095.900)b] entspricht dem öffentlichen Signaturprüfchlüssel für das CV-Zertifikat, wie er beispielsweise aus einem CVC-Root-CA entnehmbar ist.
 - b. pointInTimeaus [gemSpec_COS] entspricht der lokalen, aktuellen Systemzeit.
3. Der Use Case wird mit der Fehlermeldung InvalidEndEntityCvc beendet, wenn eine Prüfung des CV-Zertifikats gemäß [gemSpec_COS#(N095.900)b] aus der Datei EF.C.eGK.AUT_CVC.E256 mit einem Fehler endet, dabei gilt:
 - a. affectedObjectaus [gemSpec_COS#(N095.900)b] entspricht dem öffentlichen Signaturprüfchlüssel für das CV-Zertifikat, wie er beispielsweise aus dem CV-Zertifikat aus der Datei EF.C.CA.CS.E256 entnehmbar ist.
 - b. pointInTimeaus [gemSpec_COS] entspricht der lokalen, aktuellen Systemzeit.
4. Aus dem Antwortdatenfeld der vorletzten Antwortnachricht wird ein X.509-Zertifikat erzeugt. Der Use Case wird mit der Fehlermeldung InvalidX509 beendet, wenn die Prüfung dieses Zertifikates gemäß [A_28395*] fehlschlägt.
5. Dem Antwortdatenfeld der letzten Antwortnachricht wird eine Signatur entnommen. Die Signatur wird mit dem öffentlichen Schlüssel aus dem CV-Zertifikat aus der Datei

EF.C.eGK.AUT_CVC.E256 gegen das Token aus [A_28391] Punkt 6.a geprüft. Der Use Case bricht mit der Fehlermeldung InvalidAuthentication ab, wenn diese Signaturprüfung fehlschlägt.

6. Die eGK-Hash-Datenbank wird befragt, ob das CV-Zertifikat aus der Datei EF.C.eGK.AUT_CVC.E256 sowie das X.509-Zertifikat aus ein und derselben eGK stammen (siehe Funktion "check(cvc, x509, "T=CL") in [A_28400*]). Falls die Funktion mit:
 - a. "unknown" antwortet, dann wird der Use Case mit der Fehlermeldung UnknownCertificates beendet.
 - b. "mismatch" antwortet, dann wird der Use Case mit der Fehlermeldung InvalidCertificatePairContactless beendet.

[<=, ,]

Neues Kapitel: 5.4.2.3 eGK G3

Dieses Kapitel behandelt die kontaktlose Kommunikation einer eGK der Generation 3, so wie es Stand März 2025 geplant ist.

Im Gutfall schickt der sektorale IDP nach dem in [A_28387*] beschriebenen Szenario ein weiteres Szenario mit einer Reihe von Kommando-APDU an die Karte:

analog PoPP A_27022

A_28393 -Sektoraler IDP: Szenario SceAuthG3

Der sektorale IDP MUSS bei einer Kommunikation mit einer eGK G3 im zweiten Szenario folgende Liste verwenden:

1. Element:
 - a. CommandApdu: '00 a4 040c 0a a0000000167455349474e'
 - b. ExpectedStatusWord: {'9000', '6281'}
2. Element:
 - a. CommandApdu: '00 22 41B6 06 (84-01-91) || (80-01-00)'
 - b. ExpectedStatusWord: {'9000', '6281'}
3. Element:
 - a. CommandApdu: '00 b0 9100 00 0000'
 - b. ExpectedStatusWord: {'9000', '6281'}
4. Element:
 - a. CommandApdu: '00 2a 9e9a xy hashChallenge 00'
 - b. ExpectedStatusWord: {'9000', '6281'}

Hinweis 1: Das erste Listenelement selektiert das Verzeichnis DF.ESIGN.

Hinweis 2: Das zweite Listenelement ist ein MSE Set Kommando gemäß [gemSpec_COS#(N102.900)] zur Selektion des privaten Schlüssels PrK.CH.AutU.E256 für den Algorithmus signECDSA.

Hinweis 3: Das dritte Listenelement liest das X.509-Zertifikat aus der Datei EF.CH.AutU.E256.

Hinweis 4: Das vierte Listenelement ist ein PSO Compute Digital Signature Kommando gemäß [gemSpec_COS#(N087.500)] wobei das Kommandodatenfeld einen Hashwert über eine Challenge enthält. Die zugehörige Antwort-APDU wird im Erfolgsfall eine Signatur zur Challenge enthalten. [<=, ,]

analog PoPP A_27023

A_28394 -Sektoraler IDP: Auswertung SceAuthG3

Der sektorale IDP MUSS die Kartenantworten auf das Scenario SceAuthG3 wie folgt auswerten:

1. Der Use Case wird mit der Fehlermeldung UnexpectedStatusWordSceAuthG3 beendet, wenn mindestens eine Kartenantwort ein unerwartetes Statuswort enthält.
2. Aus dem Antwortdatenfeld der vorletzten Antwortnachricht wird ein X.509-Zertifikat erzeugt.
3. Die Signatur im Antwortdatenfeld der letzten Antwortnachricht wird mit dem Signaturprüfchlüssel aus dem X.509-Zertifikat geprüft. Der Use Case wird mit der Fehlermeldung InvalidAuthentication beendet, wenn die Signaturprüfung oder die Prüfung des X.509-Zertifikats gemäß [A_28395*] fehlschlägt.

[<=, ,]

Neues Kapitel: 5.4.2.4 Prüfung des X.509-Zertifikates einer eGK

In [A_28392*] und [A_28394*] wird gefordert, dass der sektorale IDP das aus einer eGK ausgelesene X.509-Zertifikat zu prüfen hat. Dies geschieht wie folgt:

analog PoPP A_27130

A_28395 -Sektoraler IDP: Prüfen von X.509-Zertifikaten einer eGK

Der sektorale IDP MUSS das aus einer eGK ausgelesene X.509-Zertifikat in Anlehnung an [gemSpec_PKI#TUC_PKI_018] mit den folgenden Eingangsparametern prüfen:

1. x509: das zu prüfende Zertifikat,
2. referenceTime: die aktuelle Systemzeit als Referenzzeitpunkt,
3. policyList = {policyIdentifier = <oid_egk_aut>},
4. keyUsage = digitalSignature,
5. extendedKeyUsage = {keyPurposeld = id-kp-clientAuth},
6. ocspGracePeriod = default ,
7. offlineModus = nein,
8. beigefügteOcspResponse: entfällt,
9. timeoutParameter = 10 Sekunden,
10. TOLERATE_OCSP_FAILURE = false,
11. prüfModus = OCSP.

Zu beachten ist A_23225*, wobei die Gültigkeitsdauer D auf 12 Stunden zu setzen ist.
[<=]

Hinweis: Der OCSP-Responder des TSP-eGK ist im Internet erreichbar. Die Adresse des OCSP-Responder ist dem Authority Information Access (AIA) des eGK-Zertifikats zu entnehmen. [<=, ,]

Neues Kapitel: 5.4.2.5 eGK-Handling Fehlercodes

In den vorherigen Kapiteln zum eGK-Handling werden diverse Fehlermeldungen definiert. Dabei ist es das Ziel für jede Stelle, an der ein Fehler detektierbar ist, eine eigene Fehlermeldung zu definieren, für den Fall, dass es an der Außenschnittstelle relevant ist. Derart aussagekräftige Fehlermeldungen unterstützen in der Entwicklungs- und

Testphase das Debugging. Für den realen Betrieb ist zweifelhaft, ob aussagekräftige Fehlermeldungen sinnvoll sind. Mitunter wird gewünscht einem Angreifer nicht zu viel darüber zu verraten, an welcher Stelle genau sein Angriff scheiterte. Konkret auf das eGK-Handling bezogen ist es aus Sicht menschlicher Nutzer eher so, dass eine eGK sich eignet um sich am sektoralen IDP zu authentifizieren, oder eben nicht. Die Tabelle in diesem Kapitel weist jeder (internen) Fehlermeldung des PoPP-Service eine Fehlermeldung zu, die an der Außenschnittstelle am Interface [[OpenAPI.yaml](#) wird nachgereicht] sichtbar sind.

analog PoPP A_27049

A_28396 -Sektoraler IDP: Mapping von Smartcard Fehlercodes

Der sektorale IDP MUSS interne Fehlermeldungen während des eGK-Handling auf folgende an Außenschnittstellen sichtbare Fehlermeldungen abbilden:

Tabelle 3: Fehlermeldungen eGK-Handling

Interne Fehlermeldung	Fehlermeldung Außenschnittstelle
InvalidAuthentication	ErrorEgkHandling
InvalidCaCvc	ErrorEgkHandling
InvalidCertificatePairContactless	ErrorEgkHandling
InvalidCertificatePairT1	ErrorEgkBlocked
InvalidEndEntityCvc	ErrorEgkHandling
InvalidPiObjectSystem	ErrorEgkHandling
InvalidPtvObjectSystem	ErrorEgkHandling
InvalidX509	ErrorEgkHandling
UnexpectedStatusWordSceAuthG2	ErrorEgkHandling
UnexpectedStatusWordSceAuthG3	ErrorEgkHandling
UnexpectedStatusWordSceOpenEgk	ErrorEgkHandling
UnexpectedStatusWordSceReadCvc	ErrorEgkHandling
UnexpectedStatusWordSceReadX509	ErrorEgkHandling
UnexpectedStatusWordSceTC1	ErrorEgkHandling
UnknownCertificates	WarningUnknownCertificates

[<=, ,]

Neues Kapitel: 5.4.2.6 eGK-Hash DB

Übersicht zu den weiteren Unterabschnitten:

1. "Einleitung und Mengengerüst": die eGK-Hash-Datenbank wird auf hoher Abstraktionsebene beschrieben
2. "Use Cases im laufenden Betrieb": Analyse der Fälle, die im laufenden Betrieb auftreten, das ist die Grundlage für die Spezifikation der "check(. . .)" Funktion in [A_28401*]
3. "Definition von Begriffen zur Wahrscheinlichkeit": Definition von Begriffen, die in Folgeabschnitten verwendet werden
4. "Use Cases zur Befüllung durch Kostenträger": Analyse der Fälle, die beim Befüllen durch Kostenträger (oder deren Dienstleister) auftreten, das ist die Grundlage für die Spezifikation der "import(. . .)" Methode in [A_28402*]
5. "Weitere Anforderungen an die eGK-Hash-Datenbank": unter anderem Spezifikation der "check(. . .)" Funktion und der "import(. . .)" Methode, basierend auf der Analyse in vorherigen Abschnitten

Neues Kapitel: 5.4.2.6.1 Einleitung und MengengerüsteGK-Hash DB

Die eGK-Hash-Datenbank im sektoralen IDP beantwortet die Frage: "Stammt ein vorgelegtes CV-Zertifikat und ein vorgelegtes X.509 AUT-Zertifikat aus ein und derselben eGK?"

So eine eGK-Hash-Datenbank wird im sektoralen IDP für eGK der Generation 2.x benötigt. Die einfachste Art der technischen Umsetzung wäre eine (mathematische) Funktion, die jedem CV-Zertifikat genau ein AUT-Zertifikat zuordnet. Softwaretechnisch wäre das eine Tabelle (in Java ein Map<CVC, AUT>). In dem Fall enthielte die Tabelle personenbezogene Daten, was weder datensparsam noch datenschutzfreundlich wäre. Stattdessen verwendet die eGK-Hash-Datenbank eine MengeegkEntries, die statt der Zertifikate unter anderem Hashwerte der Zertifikate enthält. Wenn der eGK-Hash-Datenbank dann ein Paar aus CV-Zertifikat und AUT-Zertifikat präsentiert wird, beantwortet die eGK-Hash-Datenbank letztendlich die Frage: "Ist a) dem Hashwert des CV-Zertifikats der Hashwert des AUT-Zertifikats zugeordnet und b) wenn das CV-Zertifikat unbekannt ist, ist dem Hashwert des AUT-Zertifikats nicht bereits der Hashwert eines anderen CV-Zertifikats zugeordnet?"

Überlegungen zum Mengengerüst: Es gibt (Stand März 2025) fast 75 Millionen gesetzlich Versicherte mit eGK G2.x. Es ist davon auszugehen, dass die Anzahl "nicht abgelaufener eGK" darüber liegt, weil es Versicherte gibt, die über mehr als eine "nicht abgelaufene eGK" verfügen, beispielsweise Ersatz für defekte eGK oder infolge eines Kassenwechsels. Hier wird geschätzt, dass die eGK-Hash-Datenbank so zu dimensionieren ist, dass die Mächtigkeit der Menge egkEntries bis zu 150.000.000 (150 Millionen) reicht. Zweimal 150 Millionen SHA-256 Werte beanspruchen netto 9.400 Millionen Byte, also 9,4 Gigabyte. Das ist eine Größenordnung, die für eine moderne Infrastruktur keine besonderen Ansprüche stellt.

Überlegungen zur Befüllung der MengeegkEntries: Ein neues Element wird der MengeegkEntries hinzugefügt, wenn ein KTR (oder dessen Dienstleister) das neue Element dem sektoralen IDP mitteilt. Bei der Anlieferung von Elementen für die MengeegkEntries verwendet der KTR (oder dessen Dienstleister) eine mTLS-Verbindung und übermittelt die Elemente als signierte Nachricht.

Überlegungen zum Entfernen von Elementen aus egkEntries: Es ist vorgesehen, dass KTR (oder deren Dienstleister) in der Lage sind das Entfernen eines Elementes aus egkEntries zu veranlassen (remove). Zusätzlich ist vorgesehen, dass jedem Element ein Verfallsdatum zugeordnet ist, welches sich aus dem Element "notAfter" aus dem AUT-Zertifikat ergibt. Die eGK-Hash-Datenbank ist damit in der Lage "abgelaufene" Elemente

aus der Menge egkEntries zu entfernen. Die eGK-Hash-Datenbank wird nicht dazu verpflichtet "abgelaufene" Elemente aus der Menge egkEntries zu entfernen. Falls die eGK-Hash-Datenbank in der Lage ist auch mit einer sehr großen Anzahl an Elementen in egkEntries performant umzugehen, dann ist ein Entfernen "abgelaufener" Werte auch nicht erforderlich. Die eGK-Hash-Datenbank beantwortet ja nur die Frage, ob ein vorgelegtes Paar aus einer eGK stammt, nicht aber, ob das AUT-Zertifikat des Paares noch gültig ist. Die Frage "ist ein AUT-Zertifikat noch gültig?" wird nicht von der eGK-Hash-Datenbank, sondern an anderer Stelle beantwortet (siehe Prüfung des AUT-Zertifikats in [A_28395*]).

Definition „Lieferant“: Mit „Lieferant“ ist in diesem Unterkapitel eine Instanz gemeint, die berechtigt ist dem sektoralen IDP neue Elemente für die eGK-Hash-Datenbank zu übermitteln. Es ist möglich, dass die Rolle des „Lieferanten“ von einem Kostenträger selbst wahrgenommen wird. Vermutlich wird es Kostenträger geben, welche das Einliefern von neuen Elementen an ihre Dienstleister delegieren. „Lieferant“ ist im Folgenden damit eine verkürzte Form von „Kostenträger (oder deren Dienstleister)“.

Definition listImportClients: Die Liste listImportClients enthält Identitäten, die der sektorale IDP akzeptiert, wenn diese als Client eine mTLS-Verbindung aufbauen zum Zweck des Imports von Elementen in die eGK-Hash-Datenbank.

Definition listSignatureVerificationKeys: Die Liste listSignatureVerificationKeys enthält Identitäten, die der sektorale IDP akzeptiert, wenn diese signierte Nachrichten zum Zweck des Imports von Elementen an die eGK-Hash-Datenbank übermitteln.

analog PoPP A_27043

A_28397 -Sektoraler IDP: Mindestanzahl von Element-Lieferanten

Der sektorale IDP MUSS für listImportClients und listSignatureVerificationKeys mindestens 20 Elemente pro Liste unterstützen. [\leq][\leq , ,]

Hinweis: Elemente für die Liste listImportClients erhält der Anbieter des sektoralen IDP direkt vom jeweiligen Lieferanten und die Liste listSignatureVerificationKeys wird dem Hersteller des sektoralen IDP von der gematik zur Verfügung gestellt.

Neues Kapitel: 5.4.2.6.2 Use Cases im laufenden Betrieb

Dieser Abschnitt beschreibt die möglichen Konstellationen, die im sektoralen IDP bei der Überprüfung von CV-Zertifikaten und AUT-Zertifikaten mittels der eGK-Hash-Datenbank auftreten, also im Rahmen der "check(. . .)" Funktion nach [A_28401*]. Es gilt folgende Nomenklatur:

1. Für das genutzte Übertragungsprotokoll zur eGK sind zwei Werte möglich:
 - a. "T=1": kontaktbehaftete Übertragung mit dem Protokoll T=1 aus ISO/IEC 7816-3.
 - b. Nicht "T=1", also "T=CL": kontaktlose Übertragung mit einem Protokoll aus der ISO/IEC 14443 Serie.
2. Für das CV-Zertifikat, welches dem sektoralen übermittelt wird, sind zwei Werte möglich:
 - a. "CVC bekannt": Das übermittelte CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "CVC bekannt" also "CVC unbekannt": Das übermittelte CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
3. Für das AUT-Zertifikat, welches dem sektoralen IDP übermittelt wird, sind zwei Werte möglich:

- a. "AUT bekannt": Das übermittelte AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
- b. Nicht "AUT bekannt" also "AUT unbekannt": Das übermittelte AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
4. Für das vorgelegte Paar aus CV-Zertifikat und AUT-Zertifikat sind zwei Werte möglich:
 - a. "match": Die eGK-Hash-Datenbank bestätigt, dass das vorgelegte CV-Zertifikat zum vorgelegten AUT-Zertifikat gehört.
 - b. Nicht "match" also "mismatch": Die eGK-Hash-Datenbank ist nicht in der Lage zu bestätigen, dass das vorgelegte CV-Zertifikat zum vorgelegten AUT-Zertifikat gehört. Folgende Ursachen sind möglich:
 - i. Felder 3 und 4: Weder das CV-Zertifikat noch das AUT-Zertifikat sind der eGK-Hashdatenbank bekannt.
 - ii. Felder 7 und 8: Nur das CV-Zertifikat ist der eGK-Hash-Datenbank bekannt.
 - iii. Felder 11 und 12: Sowohl das CV-Zertifikat als auch das AUT-Zertifikat sind der eGK-Hash-Datenbank bekannt, aber sie gehören nicht zusammen.
 - iv. Felder 15 und 16: Nur das AUT-Zertifikat ist der eGK-Hash-Datenbank bekannt.
5. Des Weiteren speichert die eGK-Hash-Datenbank zu jedem Element einen Zustand, der einen der folgenden Werte annimmt:
 - a. "imported": Das Element wurde mittels "import(. . .)" Methode in die eGK-Hash-Datenbank aufgenommen.
 - b. "ad hoc": Das Element wurde mittels "check(...)" Methode in die eGK-Hash-Datenbank aufgenommen.
 - c. "blocked": Der PoPP-Service stellt keine PoPP-Token aus, wenn ein vorgelegtes CV-Zertifikat oder ein vorgelegtes AUT-Zertifikat in einem Element enthalten sind, dessen Zustand "blocked" ist.

Aus der obigen Liste folgt, dass fünf Variablen zu berücksichtigen sind und die fünfte (Zustand des Elementes) keine boolesche Variable ist. Leider sind Karnaugh-Veitch-Diagramme mit mehr als vier Variablen unübersichtlich und Karnaugh-Veitch-Diagramme berücksichtigen lediglich boolesche Variablen. Zwecks Komplexitätsreduktion wird das System deshalb wie folgt vereinfacht:

1. Für alle Felder "CVC unbekannt" und "AUT unbekannt" liegt kein Element vor. Deshalb liegt auch kein Zustand vor. Daraus folgt, dass der Zustand "blocked" für solche Felder keine Rolle spielt.
2. Für alle übrigen Felder in denen der Zustand des Elementes "blocked" ist generiert der PoPP-Service eine Fehlermeldung.
3. Fazit: Der Zustand "blocked" eines Elementes ist für die weitere Betrachtung im Karnaugh-Veitch-Diagramm irrelevant. Deshalb sind hier lediglich binäre (boolesche) Variablen relevant. Zudem ist der Zustand des Elements nur für wenige Felder relevant, die dann waagerecht unterteilt werden.

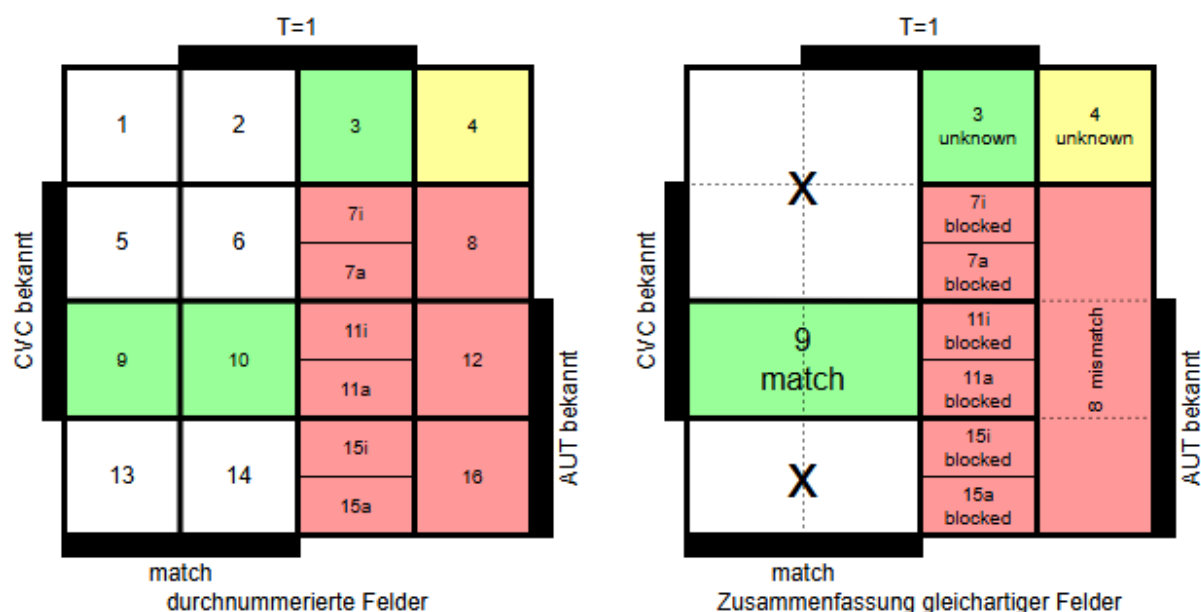


Abbildung 2 : Karnaugh-Veitch Diagramm zur "check"-Funktion

Einträge in der eGK-Hash-Datenbank werden wie folgt dargestellt:

1. **cvcX** bezeichnet den Hashwert eines bestimmten CV-Zertifikates "X".
2. **autY** bezeichnet den Hashwert eines bestimmten AUT-Zertifikates "Y".
3. **{cvcX, autY, imported}** bezeichnet einen Eintrag in der eGK-Hash-Datenbank, der durch den Import gemäß [A_28402*] entstand.
4. **{cvcX, autY, adHoc}** bezeichnet einen Eintrag in der eGK-Hash-Datenbank, der durch "trust on first (contact based) use" gemäß [A_28401*] entstand.
5. **{cvcX, autY, blocked}** bezeichnet einen Eintrag in der eGK-Hash-Datenbank, der nicht zur Erzeugung eines PoPP-Token nutzbar ist.

Für das zugehörige Karnaugh-Veitch-Diagramm gilt:

1. **Felder 1, 2, 5, 6, 13, 14:** Es gibt sechs mit "x" gekennzeichnete "don't care" Felder, weil es unmöglich ist, dass "match" zutrifft, wenn "CVC unbekannt" oder "AUT unbekannt" ist.
2. **Feld 3:** Bei kontaktbehafteter Kommunikation ist "CVC unbekannt" und "AUT unbekannt". Gemäß "trust on first (contact based) use" wird hier "ad hoc" ein neuer Eintrag zur eGK-Hash-Datenbank hinzugefügt: {cvcX, autY, adHoc}.
3. **Feld 4:** Bei kontaktloser Kommunikation ist "CVC unbekannt" und "AUT unbekannt". Der sektorale IDP ist nicht in der Lage sicher zu entscheiden, ob CV-Zertifikat und AUT-Zertifikat zusammengehören. Der sektorale IDP generiert eine Fehlermeldung.
4. **Feld 7i:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit "AUT unbekannt" vor und der Eintrag stammt aus einem Import. Aus "mismatch" folgt, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Da der Eintrag zum CV-Zertifikat "importiert" wurde ist sicher auszuschließen, dass in der Vergangenheit missbräuchlich ID_TOKEN vom sektoralen IDP ausgestellt wurden. Wegen dieses Sicherheitsvorfalls wird sowohl das CV-Zertifikat als auch die AUT-Zertifikate blockiert:
 - a. Vorgelegt werde das Paar (cvcX, autY).

- b. Zustand vorher:
 - i. {cvcX, autX, imported}
 - ii. "kein Eintrag für autY"
 - c. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcX, autY, blocked}
5. **Feld 7a:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit "AUT unbekannt" vor und der Datenbankeintrag stammt aus einem "ad hoc" Vorgang. Vorgelegt werde das Paar (cvcX, autY). Der Datenbankeintrag laute vorher: {cvcX, autZ, adHoc}. Der sektorale IDP ist in diesem Fall nicht in der Lage zu entscheiden, ob cvcX zu autY oder zu autZ oder zu einem anderen AUT-Zertifikat gehört. Es ist möglich, dass in der Vergangenheit missbräuchlich ID_TOKEN dazu von einem sektoralen IDP ausgestellt wurden. Es ist sicher, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Wegen dieses Sicherheitsvorfalls werden das CV-Zertifikat und die AUT-Zertifikate blockiert.
- a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. "kein Eintrag zu autY"
 - ii. {cvcX, autZ, adHoc}
 - c. Zustand hinterher:
 - i. {cvcX, autY, blocked}
 - ii. {cvcX, autZ, blocked}
6. **Felder 8, 12, 16:** Bei kontaktloser Kommunikation gibt es ein "mismatch". Neben dem unter Feld 7 beschriebenen Vorfall ist es hier auch möglich, dass ein Angreifer zwei verschiedene eGK dem sektoralen IDP präsentiert: Eine eGK für Authentisierung mit CV-Zertifikat und eine andere aus der das AUT-Zertifikat ausgelesen wird. Der sektorale IDP generiert eine Fehlermeldung. Der Inhalt der eGK-Hash-Datenbank wird nicht verändert.
7. **Felder 9, 10:** Bei "CVC bekannt" und "AUT bekannt" liegt ein "match" vor. Das ist der Gutfall, der zur Ausstellung eines ID_TOKEN durch den sektoralen IDP führt (egal ob der Zustand des CV-Zertifikates "imported" oder "ad Hoc" ist).
8. **Feld 11i:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit "AUT bekannt" vor und der Eintrag stammt aus einem Import. Aus "mismatch" folgt, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Da der Eintrag zum CV-Zertifikat "importiert" wurde, ist sicher auszuschließen, dass in der Vergangenheit missbräuchlich ID_TOKEN dazu ausgestellt wurden. Wegen dieses Sicherheitsvorfalls werden sowohl die CV-Zertifikate als auch die AUT-Zertifikate blockiert:
- a. Vorgelegt werde das Paar (cvcX, autY).
 - b. Zustand vorher:
 - i. {cvcX, autX, imported}
 - ii. {cvcY, autY, *=egal}
 - c. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcY, autY, blocked}

9. **Feld 11a:** Bei kontaktbehafteter Kommunikation und "CVC bekannt" liegt ein "mismatch" mit „AUT bekannt“ vor und der Datenbankeintrag stammt aus einem "ad hoc" Vorgang. Vorgelegt werde das Paar (cvcX, autY). Der Datenbankeintrag laute vorher: {cvcX, autZ, adHoc}. Der sektorale IDP ist in diesem Fall nicht in der Lage zu entscheiden, ob cvcX zu autY oder zu autZ oder zu einem anderen AUT-Zertifikat gehört. Es ist möglich, dass in der Vergangenheit missbräuchlich ID_TOKEN dazu von sektoralen IDPs ausgestellt wurden. Es ist sicher, dass der private Schlüssel zum CV-Zertifikat nicht wie vorgesehen verwendet wird. Wegen dieses Sicherheitsvorfalls werden sowohl die CV-Zertifikate als auch die AUT-Zertifikate blockiert:
- Vorgelegt werde das Paar (cvcX, autY).
 - Zustand vorher:
 - {cvcX, autZ, adHoc}
 - {cvcY, autY, *=egal}
 - Zustand hinterher:
 - {cvcX, autZ, blocked}
 - {cvcY, autY, blocked}
10. **Feld 15i:** Bei kontaktbehafteter Kommunikation und "CVC unbekannt" gibt es ein "mismatch" und der Datenbankeintrag zum vorgelegten AUT-Zertifikat wurde importiert. Da der Eintrag zum AUT-Zertifikat "importiert" wurde ist sicher auszuschließen, dass in der Vergangenheit missbräuchlich ID_TOKEN ausgestellt wurden. Wegen dieses Sicherheitsvorfalls wird sowohl das vorgelegte CV-Zertifikat als auch das vorgelegte AUT-Zertifikat blockiert:
- Vorgelegt werde das Paar (cvcX, autY).
 - Zustand vorher:
 - "kein Eintrag zu cvcX"
 - {cvcY, autY, imported}
 - Zustand nachher:
 - {cvcX, autY, blocked}
 - {cvcY, autY, blocked}
11. **Feld 15a:** Bei kontaktbehafteter Kommunikation und "CVC unbekannt" gibt es ein "mismatch" und der Datenbankeintrag zum vorgelegten AUT-Zertifikat stammt aus einem "ad hoc" Vorgang. Vorgelegt werde das Paar (cvcX, autY). Der Datenbankeintrag laute vorher: {cvcZ, autY, adHoc}. Der sektorale IDP ist in diesem Fall nicht in der Lage zu entscheiden, ob autY zu cvcX oder cvcZ oder zu einem anderen CV-Zertifikat gehört. Es ist möglich, dass in der Vergangenheit missbräuchlich ID_TOKEN dazu von einem sektoralen IDP ausgestellt wurden. Es ist sicher, dass der private Schlüssel zu wenigstens einem der beteiligten CV-Zertifikate nicht wie vorgesehen verwendet wird. Wegen dieses Sicherheitsvorfalls werden beide Einträge blockiert.
- Vorgelegt werde das Paar (cvcX, autY).
 - Zustand vorher:
 - "kein Eintrag für cvcX".
 - {cvcZ, autY, adHoc}.
 - Zustand hinterher:
 - {cvcX, autY, blocked}

- ii. {cvcZ, autY, blocked}

Neues Kapitel: 5.4.2.6.3 Definition von Begriffen zur Wahrscheinlichkeit

Hier werden einige Begriffe zu Wahrscheinlichkeiten definiert, die in Folgekapiteln verwendet werden. Die Begriffe sind nach Wahrscheinlichkeiten von "sicher" bis "unmöglich" sortiert.

1. **sicher:** Ein Ereignis tritt mit einer Wahrscheinlichkeit von eins ein.
Beispiel: Eine Urne enthält nur rote Kugeln. Die Wahrscheinlichkeit aus dieser Urne eine rote Kugel zu ziehen ist eins.
2. **extrem wahrscheinlich:** Die Eintrittswahrscheinlichkeit ist fast eins. Theoretisch ist es möglich, dass das Ereignis nicht eintritt, aber in der Praxis muss der Nichteintritt nicht betrachtet werden.
Beispiel: Zwei Zufallszahlen der Länge 128 bit, die unabhängig voneinander generiert werden, sind verschieden.
3. **sehr wahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so hoch, dass in der Praxis das Gegenereignis nicht beobachtet wird.
Beispiel: Zu einer Nachricht M wird der SHA-256 Hashwert zweimal berechnet. Beide Ergebnisse stimmen überein.
Hinweis: In [\[https://www.cs.toronto.edu/~bianca/papers/sigmetrics09.pdf\]](https://www.cs.toronto.edu/~bianca/papers/sigmetrics09.pdf) werden RAM-Lesefehler untersucht. Bei Verwendung von ECC-RAM ist es sehr wahrscheinlich, dass RAM-Lesefehler entdeckt werden. Deshalb ist es sehr wahrscheinlich, dass eine zweimalige Hashwertberechnung dasselbe Ergebnis liefert.
4. **wahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so hoch, dass das Ereignis regelmäßig beobachtet wird.
5. **unwahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so niedrig, dass das Ereignis nur selten beobachtet wird. Obwohl das Ereignis selten eintritt, müssen Systeme mit so einem Ereignis kontrolliert umgehen können, was durch Tests zu bestätigen ist.
Beispiel: Bitfehler im RAM eines Servers, beispielhafte Gegenmaßnahme ECC-RAM.
6. **sehr unwahrscheinlich:** Die Eintrittswahrscheinlichkeit ist so niedrig, dass ein System gegen so ein Ereignis nicht gehärtet werden muss.
Beispiel: Zu einer Nachricht M wird der SHA-256 Hashwert zweimal berechnet. Beide Ergebnisse sind verschieden.
7. **extrem unwahrscheinlich:** Die Eintrittswahrscheinlichkeit ist fast null. Theoretisch ist es möglich, dass das Ereignis eintritt, aber in der Praxis muss der Eintritt nicht betrachtet werden.
Beispiel: Zwei Zufallszahlen der Länge 128 bit, die unabhängig voneinander gewürfelt werden, sind gleich.
8. **unmöglich:** Ein Ereignis tritt mit einer Wahrscheinlichkeit von null ein.
Beispiel: Eine Urne enthält nur rote Kugeln. Die Wahrscheinlichkeit aus dieser Urne eine grüne Kugel zu ziehen ist null.

Neues Kapitel: 5.4.2.6.3 Use Cases zur Befüllung durch Kostenträger

Dieses Kapitel betrachtet die Befüllung der eGK-Hash-Datenbank durch Lieferanten.

Annahmen:

1. Es ist "extrem unwahrscheinlich", dass zwei verschiedene Zertifikate denselben Hashwert haben. Deshalb werden derartige Fälle hier nicht weiter betrachtet.
2. Es ist "unwahrscheinlich" (aber denkbar), dass eGKs erst im Feld eingesetzt werden und für diese ein Eintrag in die eGK-Hash-Datenbank erfolgt (per "trust on first (contact based) use") und zu einem späteren Zeitpunkt liefert ein Lieferant für

dieselbe eGK per Import einen Eintrag für die eGK-Hash-Datenbank. Daraus folgt, dass es zwar "unwahrscheinlich" aber möglich ist, dass zum Zeitpunkt des Imports bereits ein Eintrag in der eGK-Hash-Datenbank vorliegt. Dieses Kapitel betrachtet dann die dabei auftretenden Fälle und wie mit ihnen umzugehen ist.

Hinweis: Falls die Wahrscheinlichkeit des Ereignisses "eine eGK wird im Feld benutzt bevor ein Import ihrer Daten in die eGK-Hash-Datenbank" von "unwahrscheinlich" auf "sehr unwahrscheinlich" oder niedriger eingestuft wird, dann wird dieses Kapitel gegenstandslos. Derzeit ist die Annahme, dass dieses Ereignis lediglich "unwahrscheinlich" ist. Daraus folgt, dass dieses Kapitel relevant ist.

Ein neuer Eintrag {cvcX, autX, imported} wird der eGK-Hash-Datenbank neu hinzugefügt oder entfernt (remove).

1. Für das CV-Zertifikat, welches im Eintrag enthalten ist, sind zwei Werte möglich:
 - a. "CVC bekannt": Das CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "CVC bekannt" also "CVC unbekannt": Das CV-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
2. Für das AUT-Zertifikat, welches im Eintrag enthalten ist, sind zwei Werte möglich:
 - a. "AUT bekannt": Das AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank bekannt.
 - b. Nicht "AUT bekannt" also "AUT unbekannt": Das AUT-Zertifikat (genauer dessen Hashwert) ist in der eGK-Hash-Datenbank nicht bekannt.
3. Für Einträge, die das CV-Zertifikat oder das AUT-Zertifikat (oder beide) enthalten, sind zwei Werte relevant:
 - a. Wenigstens einer dieser Einträge ist im Zustand "blocked".
 - b. Keiner dieser Einträge ist "blocked" (also "unblocked").
4. Für den Vergleich zwischen dem Eintrag und einem bestehenden Eintrag in der eGK-Hash-Datenbank sind zwei Werte denkbar:
 - a. "match": Die Hashwerte im Eintrag sind identisch zu den Hashwerten in einem bestehenden Eintrag.
 - b. Nicht "match" (also "mismatch"): Die Hashwerte des Eintrags sind nicht identisch zu irgendeinem bestehenden Eintrag.

Aus der obigen Liste folgt, dass vier binäre (boolesche) Variablen zu berücksichtigen sind. Für das zugehörige Karnaugh-Veitch Diagramm gilt:



Abbildung 3 : Karnaugh-Veitch Diagramm zur "import"-Methode

1. **Felder 1, 4:** Der Fall, dass "CVC unbekannt" und "AUT unbekannt" aber "blocked" ist, kann nicht vorkommen. Deshalb sind diese Felder irrelevant (don't care).
2. **Felder 1, 2, 5, 6, 13, 14:** Der Fall, dass ein "match" vorliegt, wenn cvcX oder autX "unbekannt" sind, kann nicht vorkommen. Deshalb sind diese Felder irrelevant (don't care).
3. **Feld 3:** Weder cvcX, noch autX sind in der eGK-Hash-Datenbank enthalten.
 - a. "import": Ein neuer Eintrag wird der eGK-Hash-Datenbank hinzugefügt.
 - b. "remove": Der zu entfernende Eintrag ist nicht in der eGK-Hash-Datenbank enthalten. Die eGK-Hash-Datenbank wird nicht verändert.
4. **Felder 7, 8:** cvcX ist "bekannt" aber autX ist "unbekannt". Daraus folgt, dass der private Schlüssel des CV-Zertifikates nicht wie vorgesehen verwendet wird. In der Vergangenheit wurden missbräuchlich ID_TOKEN dazu ausgestellt. Der Eintrag wird gesperrt.
 - a. Zustand in der eGK-Hash-Datenbank vorher:
 - i. "kein Eintrag zu autX"
 - ii. {cvcX, autY, *=egal}
 - b. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcX, autY, blocked}
5. **Feld 9:** cvcX ist "bekannt" aber "blocked" und es liegt ein "match" vor. Die eGK-Hash-Datenbank wird nicht verändert.
6. **Feld 10:** Sowohl cvcX, als auch autX sind "bekannt" und auch in der eGK-Hash-Datenbank sind cvcX und autX einander zugeordnet ("match") und der Zustand von cvcX in der eGK-Hash-Datenbank ist nicht "blocked".

- a. "import": Ein neuer Eintrag wird der eGK-Hash-Datenbank hinzugefügt. Dann lassen sich folgende Fälle unterscheiden: Der Zustand von cvcX in der eGK-Hashdatenbank ist
 - i. "imported": Keine Aktion erforderlich, da die eGK-Hash-Datenbank den neuen Eintrag bereits enthält.
 - ii. "adHoc": Der Zustand des cvcX in der eGK-Hashdatenbank wird von "adHoc" auf "imported" geändert.
 - A. Zustand vorher: {cvcX, autX, adHoc}
 - B. Zustand nachher: {cvcX, autX, imported}
 - b. "remove": Der Eintrag wird aus der eGK-Hash-Datenbank entfernt.
7. **Felder 11, 12:** Sowohl cvcX, als auch autX sind "bekannt", aber in der eGK-Hash-Datenbank einander NICHT zugeordnet ("mismatch"). In der Vergangenheit wurden dazu ID_TOKEN missbräuchlich ausgestellt. Einträge werden gesperrt.
- a. Zustand vorher:
 - i. {cvcX, autY, *=egal}
 - ii. {cvcZ, autX, *=egal}
 - b. Zustand nachher:
 - i. {cvcX, autY, blocked}
 - ii. {cvcZ, autX, blocked}
8. **Felder 15, 16:** cvcX ist "unbekannt", autX ist "bekannt". Daraus folgt, dass der private Schlüssel des CV-Zertifikates nicht wie vorgesehen verwendet wird. In der Vergangenheit wurden missbräuchlich dazu ID_TOKEN ausgestellt. Der vorhandene Eintrag wird gesperrt und ein neuer Eintrag angelegt.
- a. Zustand vorher:
 - i. "kein Eintrag für cvcX"
 - ii. {cvcY, autX, *=egal}
 - b. Zustand nachher:
 - i. {cvcX, autX, blocked}
 - ii. {cvcY, autX, blocked}

analog PoPP A_27044

A_28398 -sektoraler IDP: Schnittstelle I_IDP_EHC_CertHash_Import

Der sektorale IDP MUSS über eine Schnittstelle I_IDP_EHC_CertHash_Import mit folgenden Eigenschaften verfügen:

1. Die Schnittstelle I_IDP_EHC_CertHash_Import ist aus dem Internet erreichbar.
2. Die IP-Adresse über welche die Schnittstelle I_IDP_EHC_CertHash_Import erreichbar ist wird bekanntgegeben.
3. Änderungen an der IP-Adresse für die Schnittstelle I_IDP_EHC_CertHash_Import werden mindestens 30 Tage vor der Änderung allen Inhabern der Identitäten aus der Liste listImportClients bekanntgegeben.
4. Die Schnittstelle I_IDP_EHC_CertHash_Import ist erst nach einem TLS-Handshake nutzbar. Der TLS-Handshake lässt ausschließlich Ciphersuiten gemäß

[gemSpec_Krypt] zu. Der TLS-Handshake scheitert, wenn die Identität des Client nicht Element der Liste listImportClients ist.

5. Über die Schnittstelle I_IDP_EHC_CertHash_Import wird (nach dem TLS-Handshake auf Applikationsebene) eine oder mehrere signierte Nachrichten mit Einträgen für die eGK-Hash-Datenbank übertragen (siehe [A_28400*] signedMessage).
6. Ein über die Schnittstelle importierter Eintrag ist spätestens nach 36 Stunden in der Menge egkEntries der eGK-Hash-Datenbank enthalten, sofern er erfolgreich geprüft wurde. [\leq]

[\leq , ,]

analog PoPP A_27045

A_28399 -sektoraler IDP: Eintrag Lieferant

Ein Lieferant MUSS die Schnittstelle I_IDP_EHC_CertHash_Import wie folgt bedienen:

1. Seine Identität für den TLS-Handshake wird dem Anbieter sektoraler IDP bekanntgegeben.
2. Seine Identität zur Prüfung von Signaturen wird der gematik bekanntgegeben.
3. Für neu produzierte eGK wird deren Eintrag spätestens 36 Stunden vor Auslieferung der eGK an den sektoralen IDP übertragen.
4. Nach einem TLS-Handshake werden bis zu einem "close_notify"-Alert nicht mehr als 20.000.000 Einträge übertragen.
5. Die zu signierende Nachricht eContent besitzt folgende ASN.1 Struktur:


```
eContent ::= SEQUENCE {
    version    INTEGER
    egkInfos   SEQUENCE OF egkInfo (SIZE(1..200000000))
}
egkInfo ::= SET {
    status     INTEGER, -- 0=import, 1=remove
    hashAut    BIT STRING,
    hashCvc    OCTET STRING,
    notAfter   UTCTime -- attribute "notAfter" from X.509
}
```
6. Die zu signierende Nachricht eContent wird DER codiert.
7. Als Versionsnummer wird der Wert 0 verwendet.
8. Die zu signierende Nachricht eContent wird mit einem gemäß [gemSpec_Krypt] zulässigen Verfahren wie in [RFC 5652#5] beschrieben signiert, wobei das Zertifikat des signierenden in die signierte Nachricht einzustellen ist.
9. Im Anschluss an den Import und als Antwort auf die signierte Nachricht eContent wird dem Importeur in derselben TLS-Session folgende Information zurückgemeldet:
 - a. Liste mit fehlerhaften Einträgen,
 - b. Liste mit ignorierten Einträgen,
 - c. Liste geblockter Einträge (weil der Eintrag bereits vorhanden war und geblockt war oder geblockt wurde). [\leq]

Hinweis: Die Beschränkung der Anzahl von Einträgen in A_28399, Punkt 5 auf 20 Millionen sorgt dafür, dass die signierte Nachricht kleiner als 2 GiByte = 2.147.483.647 Byte bleibt. Größere Nachrichten lassen sich mit Standardbibliotheken unter gängigen Programmiersprachen (etwa Java) nicht verarbeiten. Falls ein Lieferant mehr Einträge anliefern möchte, dann verwendet er mehrere signierte Nachrichten.* [\leq , ,]

Neues Kapitel: 5.4.2.6.4 Weitere Anforderungen an die eGK-Hash-Datenbank

analog PoPP A_27046

A_28400 -sektoraler IDP: eGK-Hash-Datenbank

Der sektorale IDP MUSS über eine eGK-Hash-Datenbank mit folgenden Eigenschaften verfügen:

1. Die eGK-Hash-Datenbank besitzt eine Menge egkEntries.
2. Die eGK-Hash-Datenbank ist in der Lage in der Menge egkEntries mindestens 150.000.000 Einträge zu speichern.
3. Die eGK-Hash-Datenbank besitzt eine Funktion mit der Signatur "String check(byte[] cvc, byte[] aut, String protocol)" gemäß [A_28401*].
4. Die eGK-Hash-Datenbank besitzt eine Methode mit der Signatur "void import(byte[]SignedData)" gemäß [A_28402*].
5. Wenn über die Funktionen "check(. . .)" oder "import(. . .)" mehr Einträge angeliefert werden, als in der eGK-Hash-Datenbank speicherbar sind, dann werden zusätzliche Einträge ignoriert. [\leq]

[\leq , ,]

analog PoPP A_27622

A_28401 -sektoraler IDP: eGK-Hash-Datenbank, check-Funktion

Die eGK-Hash-Datenbank MUSS eine Funktion mit der Signatur "String check(byte[] cvc, byte[] aut, String protocol)" und folgendem Verhalten besitzen:

1. Der Parameter "cvc" ist ein Bytestring, dessen Inhalt identisch ist zum Inhalt der Datei EF.C.eGK.AUT_CVC.E256 einer eGK.
2. Der Parameter "aut" ist ein Bytestring, dessen Inhalt identisch ist zum Inhalt der Datei EF.C.CH.AUT.E256 einer eGK.
3. Der Parameter "protocol" kennzeichnet wie mit der eGK kommuniziert wurde:
 - a. "T=1" für die kontaktbehaftete Kommunikation
 - b. "T=CL" für die kontaktlose Kommunikation
4. Die Funktion check(. . .) berechnet SHA-256 Hashwerte gemäß [FIPS PUB 180-4] und extrahiert das Attribut "notAfter" aus dem AUT-Zertifikat, es gilt:
 - a. hashCvc = SHA-256(cvc)
 - b. hashAut = SHA-256(aut)
 - c. notAfter = Attribut "notAfter" aus dem AUT-Zertifikat.
5. Die Funktion check(. . .) führt folgende Schritte aus:
 - a. Schritt 1, "**blocked**", (alle Felder): FallsegkEntries einen Eintrag für hashCvc enthält und dessen Zustand ist "blocked", oder einen Eintrag für hashAut enthält und dessen Zustand ist „blocked“, dann gibt die Funktion den Wert "blocked" zurück.
 - b. Schritt 2, "**match**", (Felder 9, 10): Falls egkEntries einen Eintrag für hashCvc enthält und dieser Eintrag enthält hashAut, dann gibt die Funktion den Wert "match" zurück.
 - c. Schritt 3, der Parameter "protocol" zeigt eine kontaktbehaftet angebundene eGK an. Dann gilt:
 - i. Schritt 3.1, "**unknown**", (Feld 3): FallsegkEntries weder hashCvc noch hashAut enthält, dann gibt die Funktion "unknown" zurück. Zusätzlich wird ein neuer Eintrag in egkEntries erzeugt:
{hashCvc, hashAut, adHoc}

- ii. Schritt 3.2, "**blocked**", (Feld 7): Falls egkEntries den Wert hashCvc enthält aber dessen Eintrag enthält nicht hashAut (sondern autX), dann gibt die Funktion "blocked" zurück.
 - A. Zustand vorher:
 - I. {hashCvc, autX, *=egal}
 - II. "kein Eintrag für hashAut"
 - B. Zustand nachher:
 - I. {hashCvc, autX, blocked}
 - II. {hashCvc, hashAut, blocked}
- iii. Schritt 3.3, "**blocked**", (Feld 11): Falls egkEntries die Werte hashCvc und hashAut enthält, aber hashCvc ist nicht hashAut zugeordnet (sondern autX, „mismatch“), dann gibt die Funktion "blocked" zurück.
 - A. Zustand vorher:
 - I. {hashCvc, autX, *=egal}
 - II. {cvcY, hashAut, *=egal}
 - B. Zustand nachher:
 - I. {hashCvc, autX, blocked}
 - II. {cvcY, hashAut, blocked}
- iv. Schritt 3.4, "**blocked**", (Feld 15): Falls egkEntries den Wert hashCvc nicht enthält, aber den Wert hashAut enthält (innerhalb eines Eintrags zu cvcY), dann gibt die Funktion "blocked" zurück.
 - A. Zustand vorher:
 - I. "kein Eintrag für hashCvc"
 - II. {cvcY, hashAut, *=egal}
 - B. Zustand nachher:
 - I. {hashCvc, hashAut, blocked}
 - II. {cvcY, hashAut, blocked}
- d. Schritt 4, "**unknown**", "**mismatch**", (Felder 4, 8, 12, 16): Der Parameter "protocol" zeigt eine kontaktlos angebundene eGK an. Dann gilt: Falls egkEntries weder hashCvc noch hashAut enthält, dann gibt die Funktion "unknown" zurück, sonst gibt die Funktion "mismatch" zurück. egkEntries wird nicht verändert.
- e. Logging innerhalb der „check(. . .)“-Funktion: Es wird ein Log-Eintrag mit hashCvc und hashAut erzeugt, falls während der Abarbeitung der „check(. . .)“-Funktion
 - i. der Wert "blocked" zurückgegeben wird.
 - ii. Einträge blockiert werden. In diesem Fall wird auch ein Kurzzeitprotokoll angelegt mit den Informationen ICCSN aus cvc und IK-Nummer aus x509. Anhand der IK-Nummer werden blockierte ICCSN spätestens am nächsten Werktag an den jeweiligen Kostenträger gemeldet. Gemeldete Informationen werden aus dem Kurzzeitprotokoll gelöscht.

[<=, ,]

analog PoPP A_27623

A_28402 -sektoraler IDP: eGK-Hash-Datenbank, import-Funktion

Die eGK-Hash-Datenbank MUSS eine Methode mit der Signatur "void import(byte[] SignedData)" und folgendem Verhalten besitzen:

1. Schritt 1: Die Methode prüft, ob der öffentliche Signaturprüf Schlüssel in SignedData in der Liste listSignatureVerificationKeys enthalten ist. Falls nicht, dann bricht die Methode ab, sonst fährt sie mit dem nächsten Schritt fort.
2. Schritt 2: Die Methode prüft die Signatur im Parameter SignedData gemäß [RFC 5652#5]. Falls die Signatur ungültig ist, dann bricht die Methode ab, sonst fährt sie mit dem nächsten Schritt fort.
3. Schritt 3: Die Methode entnimmt dem Parameter SignedData die darin enthaltene NachrichtContent.
4. Schritt 4: Die in der NachrichtContent enthaltenen Informationen beeinflussen die eGK-Hash-Datenbank wie folgt:
 - a. Falls eContent nicht die in [A_28399*] dargestellte Struktur besitzt, dann bricht die Methode ab.
 - b. Die Elemente der Liste egkInfos werden nacheinander bearbeitet. Falls ein Element egkInfo nicht die in [A_28399*] dargestellte Struktur besitzt, dann wird es übersprungen und der Zähler counterMalformedEgkInfo wird inkrementiert. Andernfalls werden die darin enthaltenen Informationen notAfter, hashCvc, hashAut und status in der eGK-Hash-Datenbank auf die in Schritt 5 beschriebene Art und Weise verarbeitet.
5. Schritt 5:
 - a. Felder 3, 15, 16: Falls hashCvc in egkEntries "unbekannt" ist und hashAut ist in egkEntries
 - i. ebenfalls "unbekannt" (Feld 3) und der Status ist
 - A. status = 0 = "import", dann wird ein neuer Eintrag erzeugt und der Zähler counterImported wird inkrementiert:
{hashCvc, hashAut, imported}
 - B. status = 1 = "remove", dann wird die eGK-Hash-Datenbank durch dieses Element egkInfo nicht verändert aber der Zähler counterRemoved wird inkrementiert.
 - ii. "bekannt" (Felder 15, 16), dann wird egkEntries wie folgt geändert und der Zähler counterBlocked wird inkrementiert:
 - A. vorher:
 - I. "kein Eintrag für hashCvc"
 - II. {cvcY, hashAut, *=egal}
 - B. nachher:
 - I. {hashCvc, hashAut, blocked}
 - II. {cvcY, hashAut, blocked}
 - b. Felder 7, 8: Falls hashCvc in egkEntries "bekannt" ist und hashAut ist "unbekannt", dann werden in egkEntries folgende Änderungen vorgenommen und der Zähler counterBlocked wird inkrementiert:
 - i. vorher:
 - A. "kein Eintrag zu hashAut"
 - B. {hashCvc, autY, *=egal}

- ii. nachher:
 - A. {hashCvc, hashAut, blocked}
 - B. {hashCvc, autY, blocked}
 - c. Feld 9: Falls hashCvc in egkEntries "bekannt" und "blocked" ist und es liegt ein "match" vor, dann wird dieses ElementegkInfo nicht weiterbearbeitet und der Zähler counterBlocked wird inkrementiert.
 - d. Feld 10: Falls hashCvc in egkEntries "bekannt" ist und es liegt ein "match" vor dann wird counterImported inkrementiert und der Status ist
 - i. status = 0 = import und der Zustand des CV-Zertifikates ist:
 - A. "imported", dann wird egkEntries nicht verändert.
 - B. "adHoc", dann wird nur dessen Zustand wie folgt geändert:
 - I. vorher: {hashCvc, hashAut, adHoc}
 - II. nachher: {hashCvc, hashAut, imported}
 - ii. status = 1 = "remove", dann werden Einträge mit hashCvc oder hashAut aus egkEntries entfernt.
 - e. Felder 11, 12: Falls sowohl hashCvc als auch hashAut in egkEntries "bekannt" sind und es liegt kein "match" vor, dann werden in egkEntries folgende Änderungen vorgenommen und der Zähler counterBlocked inkrementiert:
 - i. vorher:
 - A. {hashCvc, autY, *=egal}
 - B. {cvcZ, hashAut, *=egal}
 - ii. nachher:
 - A. {hashCvc, autY, blocked}
 - B. {cvcZ, hashAut, blocked}
6. Logging innerhalb der „import(. . .)“-Methode: Die „import(. . .)“- Methode loggt folgende Ereignisse:
- a. Es wird ein Log-Eintrag inklusive des Client-Zertifikats erzeugt, falls der TLS-Handshake fehlschlägt. Das ist dann der einzige Log-Eintrag für diesen Aufruf der "import(. . .)“-Methode.
 - b. Es wird ein Log-Eintrag inklusive des Signaturzertifikates erzeugt, wenn die Signaturprüfung fehlschlägt. Das ist dann der einzige Log-Eintrag für diesen Aufruf der "import(. . .)“-Methode.
 - c. Es wird ein Log-Eintrag mit hashCvc und hashAut erzeugt, falls während der Abarbeitung eines Elementes egkInfo Einträge blockiert werden.
 - d. Falls irgendein Zähler aus der Menge {counterBlocked, counterMalformedEgkInfo, counterImported, counterRemoved} größer als Null ist, dann wird ein Log-Eintrag inklusive der folgenden Informationen erzeugt:
 - i. Lieferant
 - ii. Wert des ZählerscounterBlocked
 - iii. Wert des Zählers counterMalformedEgkInfo
 - iv. Wert des Zählers counterImported
 - v. Wert des Zählers counterRemoved

- e. je Import (also gesammelt für alle Einträge im Paket) Absender/Client, Menge gesamt, Menge erfolgreich, Menge ignorierte, Menge fehlerhafte Einträge.

【<=, , 】

analog PoPP A_27624

A_28403 -sektoraler IDP: eGK-Hash-Datenbank, Löschen veralteter Einträge

Falls der sektorale IDP Einträge mit einem Ablaufdatum versieht und veraltete Einträge löscht, dann DARF er KEINE Einträge löschen, die als geblockt gekennzeichnet sind.

Hinweis: Es ist nicht erforderlich, dass die eGK-Hash-Datenbank für jeden Eintrag individuell ein Ablaufdatum speichert. Es ist beispielsweise möglich einem Ablaufdatum eine Menge von Einträgen zuzuordnen. Der Speicher der eGK-Hash-Datenbank enthält dieses Ablaufdatum dann nur einmal.

Hinweis: Jedes technische System hat eine Speichergrenze. Das Ignorieren von weiteren Einträgen, die sich nicht mehr speichern lassen, verhindernd undefinierte Zustände durch Speicherüberlauf.

Hinweis: Sowohl CV-Zertifikate, als auch AUT-Zertifikate verwenden aktuell SHA-256 als Hashverfahren. Deshalb ist es aus Sicherheitssicht ausreichend in der eGK-Hash-Datenbank ebenfalls SHA-256 Werte zu speichern. Insgesamt gibt es $2^{256} = 1,16 \times 10^{77}$ verschiedene SHA-256 Hashwerte. Angenommen jedem dieser Hashwerte wird ein Volumen zugeordnet, wie es ein Coronavirus einnimmt, dann käme im Mittel pro Würfel mit einer Kantenlänge von zwei Lichtjahren ein Coronavirus. Das bedeutet, dass die eGK-Hash-Datenbank im Vergleich zu allen möglichen Werten so dünn besetzt ist, dass es für einen Angreifer praktisch unmöglich ist ein Paar aus gültigem CV-Zertifikat und gültigem AUT-Zertifikat zu finden, welches nicht aus ein und derselben eGK stammt, aber trotzdem von der eGK-Hash-Datenbank eine "ja"-Antwort bekommt.

Hinweis: Die Codierung von SignedData wird derzeit mit den Kostenträgern und deren Dienstleistern abgestimmt.【<=, , 】

analog PoPP A_27201

A_28404 -sektoraler IDP: eGK-Hash-Datenbank Aspekte für Produktgutachten

Der Hersteller des sektoralen IDP MUSS die Umsetzung von:

1. Schritt 1 und Schritt 2 der import-Funktion aus [A_28402*](Signaturprüfung und Abbruch im Fehlerfall),
2. alle Schritte der check-Funktion aus [A_28401*] und der import-Funktion aus [A_28402*], die zum Status bzw. zum Zustand "blocked" führen und
3. Schritt 4 aus [A_28398*](Client-Authentisierung gegen listImportClients im TLS-Handshake und Abbruch im Fehlerfall),

im Rahmend des Produktgutachtens prüfen lassen. 【<=, , 】

Neues Kapitel: 5.4.2.6.4 Anmerkungen zur Implementierung

Die Intention dieses Unterkapitels ist es, Hilfestellungen bei der Implementierung der eGK-Hash-Datenbank zu liefern. Dieses Unterkapitel enthält keine Anforderungen.

Am Anfang dieses Unterkapitels sei zunächst angenommen, dass der sektorale IDP und mit ihm die eGK-Hash-Datenbank in Betrieb seien. Aus Performancegründen erscheint es ratsam die eGK-Hash-Datenbank im RAM zu halten, weil Speicherzugriffe auf (volatile) RAM-Inhalte vielfach schneller ablaufen als solche auf ein persistentes Dateisystem.

Gemäß [A_28401*] „check(. . .)“-Funktion Punkt 5.c.i (Feld 3) ist es möglich im laufenden Betrieb beispielsweise durch "trust on first (contact based) use" der eGK-Hash-Datenbank neue Einträge hinzuzufügen. Zudem sind in [A_28401*] weitere Punkte enthalten, bei denen sich der Inhalt der eGK-Hash-Datenbank ändert. Deshalb erscheint es nicht hinreichend zu sein, den Inhalt der eGK-Hash-Datenbank ausschließlich im RAM vorzuhalten, damit er beispielsweise durch einen Stromausfall erhalten bleibt. Daraus folgt, dass es zur eGK-Hash-Datenbank im (volatilen) RAM auch eine persistente Variante der eGK-Hash-Datenbank gibt.

Gemäß [A_28401*] und [A_28402*] spielt die Reihenfolge in welche Einträge zur eGK-Hash-Datenbank hinzugefügt werden eine Rolle.

Szenario 1:

1. Die eGK-Hash-Datenbank enthalte einen Eintrag {cvc1, aut1, imported}.
2. Aus irgendeinem Grund werde cvc1 und damit auch aut1 blockiert.
3. Anschließend werde versucht der "check(...)"-Funktion {cvc1, aut2} zu präsentieren. Ohne die eGK-Hash-Datenbank zu ändern scheitert die "check(...)"-Funktion wegen blockiertem cvc1.
4. Anschließend werde {cvc2, aut2, "T=1"} der "check(...)"-Funktion präsentiert, was zu einem neuen Eintrag {cvc2, aut2, adHoc} führt.
5. Endzustand in Szenario 1:
 - a. {cvc1, aut1, blocked}
 - b. {cvc2, aut2, adHoc}

Szenario 2:

1. Die eGK-Hash-Datenbank enthalte einen Eintrag {cvc1, aut1, imported}.
2. Anschließend werde {cvc2, aut2, "T=1"} der "check(...)"-Funktion präsentiert, was zu einem neuen Eintrag {cvc2, aut2, adHoc} führt.
3. Anschließend werde versucht der "check(...)"-Funktion {cvc1, aut2, "T=1"} zu präsentieren. Das blockiert cvc1, cvc2, aut1 und aut2.
4. Endzustand in Szenario 2:
 - a. {cvc1, aut1, blocked}
 - b. {cvc2, aut2, blocked}

Beispielhaftes Konzept für eine Implementierung der eGK-Hash-Datenbank:

1. Nach erfolgreicher Signaturprüfung speichert die "import(...)"-Methode die "SEQUENCE OF"egkInfos zunächst zusammen mit einem Zeitstempel persistent.
2. Falls im Rahmen der "check(...)"-Funktion ein neuer Eintrag angelegt wird (Feld 3), dann wird ebenfalls mit Zeitstempel eine „SEQUENCE OF“egkInfos mit den Daten aus dem neuen Eintrag persistent gespeichert, mit status=2=adHoc.
3. Falls im Rahmen der "check(...)"-Funktion oder der "import(...)"-Methode ein oder mehr Einträge blockiert werden, dann werden alle dabei blockierten Einträge in einer „SEQUENCE OF“egkInfos mit Zeitstempel persistent gespeichert, mit status=3=blocked.

Daraus ergibt sich, dass die eGK-Hash-Datenbank zweimal vorliegt: Volatil im RAM und als durch Zeitstempel geordnete Liste von „SEQUENCE OF“ egkInfos, die persistent gespeichert sind. Der Zustand im RAM lässt sich dabei jederzeit aus den persistenten Informationen eindeutig rekonstruieren, wenn die zeitlich geordneten „SEQUENCE OF“ egkInfos nacheinander verarbeitet werden.

Der Neuaufbau des RAM-Zustandes der eGK-Hash-Datenbank ist möglicherweise (je nach Implementierung) zeitintensiv. Dann bietet es sich an, von der RAM-Version der eGK-Hash-Datenbank einen persistenten Speicherabzug zu erstellen, der sich performanter ins RAM laden lässt.

Neu - Anforderung an sekt. IDP eGK ohne PIN zu unterstützen

A_27992 -sektoraler IDP: Unterstützung Authentisierungsverfahren NFC eGK ohne PIN

Der Hersteller eines sektoralen IDP MUSS ein Authentifizierungsverfahren mittels eGK ohne PIN unterstützen, bei dem er von einer kontaktlos angebundenen eGK der Generation 2.1 ausgeht. Der sektorale IDP MUSS dazu folgende Schritte durchführen:

1. Prüfung des CV-Zertifikats der eGK.
2. Erzeugung und Versand einer Challenge (es wird eine Response und das Zertifikat C.CH.AUT.E256 empfangen).
3. Prüfung der Response.
4. Prüfung des C.CH.AUT.E256-Zertifikats.
5. Prüfung gegen die Hash-DB, ob das vorgelegte End-Entity-CV-Zertifikat und das vorgelegte Zertifikat C.CH.AUT.E256 aus ein- und derselben eGK stammen.

Schlägt einer der Schritte fehl, so wird der Prozess mit einer Fehlermeldung an das Authenticator-Modul abgebrochen.

Hinweis: Die Prüfung gegen die Hash-DB ist in A_28401* beschrieben. [≤, IDP-Sek, Sich.techn. Eignung: Produktgutachten]

Neu - Anforderung an sekt. IDP Umgang mit Hash-DB

A_28230 -sektoraler IDP: Vorhalten der Hash-DB

Anbieter von sektoralen IDPs KÖNNEN die Inhalte der zentralen Hash-DB in einem lokalen Cache halten. [≤, Anb_IDP-Sek_KTR, Sich.techn. Eignung: Anbietererklärung]

A_28231 -sektoraler IDP: Befüllung der Hash-DB

Der Anbieter sek IDP KTR MUSS sicherstellen, dass die zentrale Komponente "Hash-DB" einmal täglich bezüglich ausgegebener, abgelaufener und gesperrter eGK aktualisiert wird.

[≤, Anb_IDP-Sek_KTR, Sich.techn. Eignung: Anbietererklärung]

Neu - Anforderung an Authenticator-Modul Unterstützung Authentisierungsverfahren "eGK ohne PIN"

A_28232 -Authenticator-Modul: Unterstützung Authentisierungsverfahren "eGK ohne PIN"

Das Authenticator-Modul eines sektoralen IDP MUSS das Authentisierungsverfahren "eGK ohne PIN" unterstützen und dem Nutzer unter dieser Bezeichnung anbieten.

- per NFC-Schnittstelle des Smartphones die Verbindung zur eGK herstellen,
- das Masterfile (MF) der eGK per APDUs (Kartenkommandos) auswählen (selektieren),
- einen PACE-Kanal zwischen eGK und Authenticator-Modul aufbauen,

- falls die eGK *eine* X.509-Identität besitzt, die ohne PIN Eingabe genutzt werden kann, dann
 - a. das X.509-Zertifikat der Identität lesen,
 - b. den zugehörigen privaten Schlüssel selektieren und
 - c. das INTERNAL-AUTHENTICATE-Kommando (im Rahmen eines Challenge-Response-Verfahrens) ausführen
- falls die eGK *keine* X.509-Identität besitzt, die ohne PIN Eingabe genutzt werden kann, dann
 - d. das CV-Zertifikat der Identität lesen,
 - e. den zugehörigen privaten Schlüssel selektieren,
 - f. das INTERNAL-AUTHENTICATE-Kommando (im Rahmen eines Challenge-Response-Verfahrens) ausführen,
 - g. das Verzeichnis DF.ESIGN auswählen und
 - h. das X.509-Zertifikat C.CH.AUT.E256 der Identität auslesen.

Die signierte Challenge (3) und die ausgelesenen Zertifikate MUSS das Authenticator-Modul im Authentifizierungsablauf an den sektoralen IDP übertragen.【<=, IDP-Sek, Sich.techn. Eignung: Produktgutachten】

Verschiebung Kapitel 5.45 Authenticator-Modul für Desktop-Plattformen Anwendungen

3 Änderungen in Steckbriefen

3.1 Änderungen in gemProdT_IDP-Sek_PTV

Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments [gemProdT_IDP-Sek_PTV]. Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.

Tabelle 4: Anforderungen zur sicherheits technischen Eignung "Produktgutachten"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_23129-05 (entfällt)	Identifikation des Authentisierungsverfahren	gemSpec_IDP_Sek
A_23129-06 (neu)	Identifikation des Authentisierungsverfahren	gemSpec_IDP_Sek
A_28229 (neu)	Inhalte des ID_TOKEN nach Authentifizierung mit eGK ohne PIN	gemSpec_IDP_Sek
A_27992 (neu)	sektoraler IDP: Unterstützung Authentisierungsverfahren NFC eGK ohne PIN	gemSpec_IDP_Sek
A_28232 (neu)	Authenticator-Modul: Unterstützung Authentisierungsverfahren "eGK ohne PIN"	gemSpec_IDP_Sek

3.2 Änderungen in gemAnbT_IDP-Sek_KTR_ATV

Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments [gemAnbT_IDP-Sek_KTR_ATV].

Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.

Tabelle 5: Anforderungen zur sicherheits technischen Eignung "Anbietererklärung"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_28230 (neu)	sektoraler IDP: Vorhalten der Hash-DB	gemSpec_IDP_Sek
A_28231(neu)	sektoraler IDP: Befüllung der Hash-DB	gemSpec_IDP_Sek

Tabelle 6: Anforderungen zur sicherheits technischen Eignung "Gutachten (Anbieter)"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_28137 (neu)	Definition "gematik-ehealth-loa-normal"	gemSpec_IDP_Sek