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

1 Farbliche Markierung von Änderungen

Dieses Kapitel erläutert den Farbcode der farblichen Hinterlegung von Änderungen beim Vergleich von [gemSpec_COS] in der Version 3.12.0 vom 15.05.2019 gegenüber der aktuellen Polarionversion.

1) **blaue Hinterlegung:** Editorielle Änderungen, die sich weder auf Implementierungen durch Hersteller, noch auf Testimplementierungen auswirken.
2) **gelbe Hinterlegung:** Änderungen, die sich nicht auf die Implementierung durch Hersteller, wohl aber auf gematik Artefakte (Spezifikationen, Testimplementierung, …) auswirken.
3) **orange Hinterlegung:** Änderungen, die sich auf die Implementierung durch Hersteller und / oder die gematik Artefakte auswirken.

Es ergibt sich eine Priorisierung von niedrig=ohne über blau, gelb nach orange. Im Zweifelsfall wird für eine höhere Priorisierung, also eine stärkere Warnfarbe gewählt.

2 Änderung

1) Ort Kapitel 5.1, Funktion BitLength
   a) Die Funktion wird in eine Anforderung eingebettet.

(N000.080) K_COS

Tabelle 5 , CosT_89d: Definition der Funktion BitLength(...)

Input:	*in*	Bitstring mit beliebigem Inhalt und beliebiger Länge
Output:	*out*	Integer, Anzahl der Bits aus denen *in* besteht
Errors:	–	Keine
Notation:		*out* = BitLength(*in*)

Das COS MUSS in *out* die Anzahl Bits von *in* zurückmelden.

   b) Begründung: Die Funktion wird als Anforderung referenzierbar.

2) Ort Kapitel 5.2, Funktion OctetLength
   a) Die Funktion wird in eine Anforderung eingebettet.

(N000.090) K_COS

Tabelle 6 , CosT_f2b: Definition der Funktion OctetLength(...)

Input:	*in*	Oktettstring mit beliebigem Inhalt und beliebiger Länge, oder nicht–negative ganze Zahl
Output:	*out*	Integer, Anzahl der Oktette aus denen *in* besteht, oder Anzahl Oktette, die mindestens nötig sind, um eine nicht–negative ganze Zahl zu codieren.
Errors:	–	Keine
Notation:		*out* = OctetLength(*in*)

a. Wenn *in* ein Oktettstring ist, dann MUSS das COS in *out* die Anzahl der Oktett in *in* in zurückmelden.
b. Wenn *in* eine nicht-negative ganze Zahl ist, dann MUSS das COS in *out* die Mindestanzahl an Oktetten zurückmelden, die notwendig ist um *in* hexadezimal darzustellen.

   b) Begründung: Die Funktion wird als Anforderung referenzierbar.

3) Ort Kapitel 1.1, §3, Bullet 2

a) Der Inhalt wird ersetzt

- Kapitel 2: Optionen beschreibt und listet Funktionspakete, die nicht zwingend in allen Kartentypen vorhanden sind. ~~wird in einer späteren Version Eingangsanforderungen enthalten, welche die Basis für die normativen Aussagen in späteren Kapiteln bilden werden.~~

b) Begründung: Falsche Aussage gelöscht, tatsächlichen Inhalt des Kapitels beschrieben.

4) Ort Kapitel 1.1, §3, Bulletliste
   a) Der Liste werden weitere Punkte hinzugefügt
      - Kapitel 17: Informative Hinweise zur Sicherheitsevaluierung.
      - Kapitel 18: Vorgaben zur Performanz enthält Anforderungen an einen Performanztest.
   b) Begründung: Vervollständigung der Auflistung

5) Ort Kapitel 1.4, §2
   a) Neben der Versichertenkarte werden auch die übrigen Karten der TI genannt

      einer Versicherten- Leistungserbringer, Geräte oder sonstigen Karte finden sind

   b) Begründung: Vervollständigung der Auflistung

6) Ort CosT_9f7
   a) Die erste Tabellenzeile wird ersatzlos gestrichen

Tabelle 3, CosT_9f7: Abkürzungen und Definitionen

~~G1~~	~~Abkürzung für Generation 1, bezeichnet ältere Versionen des Dokumentes~~

   b) Begründung: Entfernen einer veralteten Bezeichnung

7) Kapitel 1.5.2
   a) Dem Kapitel wird folgender § hinzugefügt

Diese Dokumentenversion wurde mit einem neuen Tool statt mit Word erstellt. Das hat folgende Auswirkungen:

1. In der Normenreihe ISO/IEC 7816 ist es üblich Kommandonamen mit Kapitälchen darzustellen. Weil dieses Feature im Tool nicht verfügbar ist werden Kommandonamen in einem anderen Font dargestellt, beispielsweise READ BINARY, PSO Compute Digital Signature.
2. Die N-Nummern enthalten oft eine Untergliederung in Unterpunkte, Unter-Unterpunkte, etc. Die Numerierung erfolgt in Word und neuem Tool teilsweise unterschiedlich:
   a. Die erste Gliederungsstufe ist sowohl in Word, als auch im Tool alphabetisch: a, b, c, d, ...
      i. Die zweite Gliederungsstufe erfolgt numerisch mit arabischen Ziffern (Word: 1, 2, ...) bzw kleinen römischen Ziffern (Tool: i, ii, iii, iv, v, ...)
         A. Die dritte Gliederungsstufe erfolgt in Word numerisch mit kleinen römischen Ziffern (i, ii, iii, iv, v, vi, ...) und im Tool alphabetisch mit Großbuchstaben (A, B, C, ...).
            I. Die vierte Gliederungsstufe erfolgt in Word alphabetisch mit Großbuchstaben (A, B, C, ...) und im Tool numerisch mit großen römischen Ziffern (I, II, III, IV, V, VI, ...)
               1. Weitere Gliederungsebenen werden in diesem Dokument in Anforderungen nicht verwendet.
   b. Daraus folgt, dass Referenzen auf N-Nummern sich von der Word-Version hin zur Tool-Version möglicherweise unterscheiden, aber ineinander umrechenbar sind. Zudem ist erkennbar, falls in der Tool-Version die Umrechnung unterblieb. Beispiel: Eine Referenz auf (N654.321)a.7.ii.D ist eine Word-Version, die im Tool so lautet: (N654.321)a.vii.B.IV. Abgesehen von dieser anderen Art der Darstellung von N-Nummern auf tieferen Gliederungsebenen sind die N-Nummern im Tool identisch zu den N-Nummern in der Word-Version.
2. Automatische Referenzierungen wie in Word (beispielsweise siehe Abbildung 3 oder siehe Tabelle 47) sind im Tool nicht möglich. Ersatzweise werden händisch gepflegte Label nach folgendem Schema verwendet:
   a. CosA_xxx für Abbildungen,
   b. CosH_xxx für Hinweise,
   c. CosT_xxx für Tabellen
   d. wobei xxx eine dreistellige Folge hexadezimaler Ziffern ist.

   b) Begründung: Erläuterung der notwendigen Umstellungen beim Übergang von Word nach Polarion.

8) Ort Kapitel 1.5.3, §3
   a) Text umformuliert

Generell gilt, dass lediglich die Gliederungen, welche durch eine Nummer (N4711) gekennzeichnet sind, zulassungsrelevante Eigenschaften enthalten ~~SOLLEN~~ und somit im Rahmen der Zulassung getestet werden ~~SOLLEN~~. Falls dies in einem speziellen Fall nicht so ist, handelt es sich höchstwahrscheinlich um einen editorischen Fehler.

   b) Begründung: Vermeiden von RFC-Notation außerhalb von Anforderungen

9) Ort CosT_485
   a) Die Zeile für K_COS_G1 wird ersatzlos gestrichen

~~K_COS_G1~~	~~Betriebssystem einer Smartcard in der Generation 1, es ist denkbar, dass derartig gekennzeichnete Anforderungen in späteren Versionen dieses Dokumentes entfallen~~

   b) Begründung: Entfernen veralteter Textstellen

10) Ort CosT_485
   a) Der Tabelle wird eine weitere Zeile angefügt

K_TST	Komponente funktionaler Zulassungstest

   b) Begründung: An eine weitere Komponente werden Anforderungen gerichtet

11) Ort Kapitel 1.5.5, §1
   a) Die RFC-Notation wird ausführlicher erklärt

gekennzeichnet. ~~Abwandlungen~~Konjugationen von "MUSS" zu "MÜSSEN" etc. sind der Grammatik geschuldet. Falls in einem speziellen Fall die Verben müssen, dürfen, sollen, können ohne einen einzigen Großbuchstaben verwendet werden, handelt es sich höchstwahrscheinlich um einen editorischen Fehler.

   b) Begründung: Präzisierung

12) Ort Kapitel 1.5.5, §3
   a) Erläuterung zu Anpassungen von Anforderungen mit dem RFC-Wort "KANN"

In dieser Version wird das Verb "kann" in normativen Festlegungen nur noch für optionale Funktionspakete verwendet, aber nicht länger im Sinne von "zulässigen Implementierungsvarianten" oder "herstellerspezifischen Erweiterungen". Beispiel alt: (N002.200) Das COS KANN weitere Moduluslängen unterstützen oder ablehnen. Die gematik betrachtet normative Festlegungen mit dem Verb "kann" als Implementierungsvarianten und erfragt bei Herstellern welche konkrete Ausprägung gewählt wurde. Das ist im gezeigten Beispiel sinnlos. Deshalb wird die etwa in (N002.200) an einen Hersteller gerichtete Erlaubnis bestimmte Erweiterungen vorzunehmen oder zu unterlassen methodenkonform angepasst.

~~In diesem Dokument werden Aussagen mit dem Schlüsselwort KANN generell sowohl positiv als auch negativ formuliert. (N002.200) ist im Zusammenhang mit (N002.100)a dafür ein gutes Beispiel. In (N002.100)a wird eine normative Forderung erhoben, die offen lässt, ob zusätzliche Werte fakultativ verboten sind oder nicht. Diese Lücke wird durch (N002.200) geschlossen.~~

   b) Begründung: Vermeidung des RFC-Wortes "KANN" im Sinne von "zulässige Implementierungsvarianten".

13) Ort (N000.020)b
   a) Anpassungen am Text

b. Wenn das IC und das COS einer Smartcard die Option_USB_Schnittstelle

1. unterstützen, dann sind~~MÜSSEN~~ zusätzlich zu allen nicht gekennz zu erfüllen, die mit Option_USB_Schnittstelle gekennzeichnet sind.
2. nicht unterstützen, dann sind~~DÜRFEN NICHT~~ mit Option_USB_Sch für funktionale Tests ~~sein~~.

   b) Begründung: Vermeiden unterschiedlicher RFC-Kategorien in einer Anforderung

14) Ort (N000.022)b
   a) Anpassungen am Text

b. Wenn das IC und das COS einer Smartcard die Option_kontaktlose_Schnittstelle

1. unterstützen, dann sind~~MÜSSEN~~ zusätzlich zu allen nicht gekennzeichneten erfüllen, die mit Option_kontaktlose_Schnittstelle gekennzeichnet sind.
2. nicht unterstützt, dann sind~~DÜRFEN NICHT~~ mit Option_kontaktlose_Schnitts für funktionale Tests ~~sein~~.

   b) Begründung: Vermeiden unterschiedlicher RFC-Kategorien in einer Anforderung

15) Ort (N000.024)b
   a) Anpassungen am Text

b. Wenn das COS die Option_logische_Kanäle

1. unterstützt, dann sind~~MÜSSEN~~ zusätzlich z erfüllen, die mit Option_logische_Kanäle gek
2. nicht unterstützt, dann sind~~DÜRFEN NICHT~~ funktionale Tests ~~sein~~.

   b) Begründung: Vermeiden unterschiedlicher RFC-Kategorien in einer Anforderung

16) Ort (N000.026)b
   a) Anpassungen am Text

b. Wenn das COS die Option_Kryptobox

1. unterstützt, dann sind~~MÜSSEN~~ zusätzlich z erfüllen, die mit Option_Kryptobox gekennze
2. nicht unterstützt, dann sind~~DÜRFEN NICHT~~ Tests ~~sein~~.

   b) Begründung: Vermeiden unterschiedlicher RFC-Kategorien in einer Anforderung

17) Ort (N000.028)b
   a) Anpassungen am Text

b. Wenn das COS die Option_PACE_PCD

1. unterstützt, dann sind~~MÜSSEN~~ zusätzlich z erfüllen, die mit Option_PACE_PCD gekenn
2. nicht unterstützt, dann sind~~DÜRFEN NICHT~~ funktionale Tests ~~sein~~.

   b) Begründung: Vermeiden unterschiedlicher RFC-Kategorien in einer Anforderung

18) Ort CosH_84b
   a) Ergänzung

Hinweis CosH_84b: Die Option_RSA_KeyGeneration wird derzeit lediglich im Rahmen der Personalisierung von Karten der Typen HBA und SMC-B sowie bei der Umsetzung der Option_lange_Lebensdauer verwendet.

   b) Begründung: Vervollständigung der möglichen Anwendungsfälle

19) Ort Kapitel 4, §3
   a) Der letzte Satz wird ersatzlos gestrichen

Dieses Dokument gilt nicht für die Vorbereitungsphase von Applikationen oder deren Bestandteile. ~~Sie beschreibt lediglich den Zustand des Objektsystems in der Nutzungsphase.~~

   b) Begründung: Es wird nicht deutlich, wer hier mit dem Subjekt "Sie" gemeint ist. Zudem ist die Beschreibung des Objektsystems Aufgabe der Objektsystemspezifikation. Dies Aussage liefert hier keinen Mehrwert und erscheint überflüssig.

20) Ort: (N000.100)

a) Die Tabelle wird in die Anforderung integriert.
b) Der Inputparameter n wird genauer beschrieben.
c) Der Anforderungstext wird um einen Satz mit RFC-Wort ergänzt.
d) Die Anmerkung zu Schritt 2 wird in einen referenzierbaren Hinweis geändert.

(N000.100) K_COS

Tabelle 7, CosT_042: Definition der Funktion I2OS(...)

Input:	x	Integer, nicht-negative ganze Zahl
	n	Integer, nicht-negative ganze Zahl, Anzahl der Oktette in *out*, n darf null sein, dann ist *out* leer
Output:	*out*	Oktettstring der Länge n Oktett
Errors:	–	Keine *ACHTUNG: Im Unterschied zu [BSI-TR-03111#3.1.2] wird hier keine Fehlermeldung erzeugt, falls x größer gleich 256^n ist*
Notation:		*out* = I2OS(x, n)

Das Prinzip ist, die nicht-negative ganze Zahl x als Ziffernfolge zur Basis 256 zu notieren und dann nur die niedrigwertigsten n Ziffern für die Ausgabe zu verwenden. Das COS MUSS dabei folgenden Algorithmus ausführen:

a. Schritt 1: $x = 256^0 x_0 + 256^1 x_1 + 256^2 x_2 + \ldots + 256^i x_i + \ldots$

b. Schritt 2: $M_i = x_i$. |

~~Anmerkung~~*Hinweis CosH_bfc: Jede Ziffer x_i wird vorzeichenlos in einem Oktett M_i codiert.*

c. Schritt 3: $out = M_{n-1} \parallel M_{n-2} \parallel \ldots \parallel M_2 \parallel M_1 \parallel M_0$. <=

e) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

21) Ort (N000.200)
a) Die Tabelle wird in die Anforderung integriert.
b) Der Anforderungstext wird um einen Satz mit RFC-Wort ergänzt.

(N000.200) K_COS

Tabelle 8 , CosT_c6a: Definition der Funktion OS2I(...)

Input:	*in*	Oktettstring beliebiger Länge und beliebigen Inhalts
Output:	*out*	Integer, nicht–negative ganze Zahl
Errors:	–	Keine
Notation:		*out* = OS2I(*in*)

Das Prinzip ist, jedes Oktett als Ziffer zur Basis 256 einer nicht-negativen ganzen Zahl im Big-Endian-Format aufzufassen. Das COS MUSS dabei folgenden Algorithmus ausführen:

a. Schritt 1: n = OctetLength(*in*)

b. Wenn n gleich null ist,

1. dann ist *out* = 0.
2. sonst wähle *out* so, dass I2OS(*out*, n) identisch zu *in* ist. <=

c) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

22) Ort (N000.300)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Der Anforderungstext wird um einen Satz mit RFC-Wort ergänzt.

(N000.300) K_COS

Tabelle 9 , CosT_3e0: Definition der Funktion OS2P(...)

nput:	*PO*	Oktettstring, codiert einen Punkt auf einer elliptischen Kurve
	dP	Domainparameter gemäß (N008.600)
Output:	*P*	Punkt auf einer elliptischen Kurve mit den Koordinaten $P = (x, y)$
Errors:	ERROR	Wenn PO nicht im Format "uncompressed encoding" vorliegt
Notation:		P = OS2P(PO, dP)

Das COS MUSS die Decodierung von *PO* gemäß [BSI-TR-03111#3.2.1] durchführen:
a. Schritt 1: Wenn OctetLength(*PO*) ungleich (2 *dP.L* + 1) ist,
dann gebe den Fehler "ERROR" zurück und beende diesen Algorithmus.
b. Schritt 2: Teile *PO* auf gemäß:
PO == *PC* || *X* || *Y*,
1 == OctetLength(*PC*),
dp.L == OctetLength(*X*) == OctetLength(*Y*).
c. Schritt 3: Wenn *PC* ungleich '04' ist,
dann gebe den Fehler "ERROR" zurück und beende diesen Algorithmus.
d. Schritt 4: Setze $P = (x, y)$ = (OS2I(*X*) mod *dP.p*, OS2I(*Y*) mod *dP.p*).
e. Schritt 5: Wenn *P* nicht auf der durch *dP* definierten Kurve liegt,
dann gebe den Fehler "ERROR" zurück und beende diesen Algorithmus.

   c) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

23) Ort (N000.400)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Der Inputparameter n wird genauer beschrieben.
   c) Der Anforderungstext wird um einen Passus mit RFC-Wort ergänzt.

(N000.400) K_COS

Tabelle 10 , CosT_e43: Definition der Funktion P2OS(...)

Input:	*P*	Punkt auf einer elliptischen Kurve mit den Koordinaten $P = (x, y)$
	n	Integer, positive ganze Zahl, Anzahl Oktette pro Koordinate
Output:	*PO*	Oktettstring, codiert einen Punkt auf einer elliptischen Kurve
Errors:	–	
Notation:		PO = P2OS(P, n)

Das COS MUSS die Codierung von *P* gemäß [BSI-TR-03111#3.2.1] vornehmen:
PO = '04' || I2OS(*x*, *n*) || I2OS(*y*, *n*).

   d) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

24) Ort (N000.500) und (N000.600)
   a) Die Tabelle wird in die Anforderung (N000.600) integriert.
   b) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N000.500) K_COS
Das COS MUSS beim Aufruf dieser Funktion sicherstellen, dass n kleiner gleich s ist.

(N000.600) K_COS

Tabelle 11 , CosT_638: Definition der Funktion Extract_MSBit(...)

Input:	*in*	Entweder Bitstring der Länge s Bit, oder Oktettstring der Länge s Bit
	n	Integer, Anzahl der zu extrahierenden Bit
Output:	*out*	Bitstring der Länge n Bit
Errors:	–	Keine
Notation:		*out* = Extract_MSBit(*in*, *n*)

Das COS MUSS in den Bitstring *out* die n MSBit von *in* einstellen.

   c) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

25) Ort (N000.700) und (N000.800)
   a) Die Tabelle wird in die Anforderung (N000.800) integriert.
   b) Die Beschreibung des Inputparameters n wird präzisiert.
   c) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N000.700) K_COS
Das COS MUSS beim Aufruf dieser Funktion sicherstellen, dass n kleiner gleich s ist.

(N000.800) K_COS

Tabelle 12 , CosT_1ff: Definition der Funktion Extract_MSByte(...)

Input:	*in*	Oktettstring der Länge s Oktett
	n	Integer, Anzahl der zu extrahierenden Oktette~~Elemente~~
Output:	*out*	Oktettstring der Länge n Oktett
Errors:	–	Keine
Notation:		*out* = Extract_MSByte(*in*, *n*)

Das COS MUSS in den Oktettstring *out* die n MSByte von *in* einstellen.

   d) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

26) Ort (N000.900)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Die Beschreibung des Inputparameters n wird präzisiert.
   c) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N000.900) K_COS

Tabelle 13 , CosT_552: Definition der Funktion PaddingIso(...)

Input:	*in*	Oktettstring mit beliebigem Inhalt und beliebiger Länge
	n	Integer, positive ganze Zahl, gibt die Blocklänge an, auf die *out* zu padden ist
Output:	*out*	Oktettstring mit einer Anzahl Oktette, die Vielfaches von *n* ist
Errors:	–	Keine
Notation:		*out* = PaddingIso(*in*, *n*)

Das COS MUSS folgende Aktion durchführen:
a. Schritt 1: *out* = *in* || '80'.
b. Schritt 2: Wenn 0 == OctetLength(*out*) mod *n*,
dann gebe *out* zurück und beende den Algorithmus,
sonst fahre mit Schritt 3 fort.
c. Schritt 3: *out* = *out* || '00'.
d. Schritt 4: Fahre mit Schritt 2 fort.

   d) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

27) Ort (N001.000)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.
   c) Die Beschreibung in Schritt 2 wurde korrigiert.

(N001.000) K_COS

Tabelle 14 , CosT_9fb: Definition der Funktion Truncatelso(...)

Input:	*in*	Oktettstring mit beliebigem Inhalt und einer Anzahl Oktette, die Vielfaches von *n* ist
	n	Integer, gibt die Blocklänge in Oktett an, auf die gepadded wurde
Output:	*out*	Oktettstring mit beliebigem Inhalt und beliebiger Länge
Errors:	*paddingError*	Die Länge von *in* ist kein Vielfaches von *n*, oder es ist kein Bit im Oktettstring *in* gesetzt, oder der Okettstring *in* ist falsch gepadded, oder es sind zu viele Paddingbits vorhanden
Notation:		*out* = Truncatelso(*in*, *n*)

Das COS MUSS folgende Aktion durchführen:

a. Schritt 1: Setze *len* = OctetLength(*in*).
   Wenn *len* kein Vielfaches von *n* ist,
   dann breche diesen Algorithmus mit der Fehlermeldung *paddingError* ab.
b. Schritt 2: Wenn OctetLength(*in*) ~~*len*~~ gleich null ist, dann
   breche diesen Algorithmus mit der Fehlermeldung *paddingError* ab.
c. Schritt 3: Wenn LSByte von *in* den Wert '00' hat,
   dann setze *in* = Extract_MSByte(*in*, OctetLength(*in*) – 1)
   und fahre mit Schritt 2 fort.
d. Schritt 4: Wenn LSByte von *in* nicht den Wert '80' hat,
   dann breche diesen Algorithmus mit der Fehlermeldung *paddingError* ab.
e. Schritt 5: *out* = Extract_MSByte(*in*, OctetLength(*in*) – 1)
f. Schritt 6: Wenn (*len* – OctetLength(*out*)) größer als *n* ist,
   dann breche diesen Algorithmus mit der Fehlermeldung *paddingError* ab.

   d) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

28) Ort (N001.100)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N001.100) K_COS

Tabelle 15 , CosT_24c: Definition der Funktion MGF(...)

Input:	Z	Bitstring mit beliebigem Inhalt und beliebiger Länge
	L_N	Integer, gibt die Bitlänge von N an
	i	Integer, Startwert der Iteration
Output:	N	Bitstring, Ergebnis der Mask Generation Function
Errors:	*error*	Gemäß [ISO/IEC 9796-2#C.3.2] Punkt 1 ist ein Fehler zu werfen, falls Z zu lang, oder L_N zu groß ist. *Hinweis CosH_ff6: Dieser Fehler ist für Smartcards derzeit nicht praxisrelevant, da er erst dann auftritt, wenn Z oder N oberhalb mehrerer Gigabyte liegen.*
Notation:		N = MGF(Z, L_N , i)

Das COS MUSS bei der Berechung von N aus Z, L_N und i folgenden Algorithmus ausführen:

a. Schritt 1: Setze n = '', (Anmerkung: Leerer Oktettstring).
b. Schritt 2: Setze n = n || SHA_256(BS2OS(Z) || I2OS(i, 4)).
c. Schritt 3: Wenn die Hash-Operation mit einem Fehler terminierte,
dann breche diesen Algorithmus mit der Fehlermeldung *error* ab.
d. Schritt 4: Setze i = i + 1.
e. Schritt 5: Wenn i größer gleich 2^{32} = '1 0000 0000' ist,
dann breche diesen Algorithmus mit der Fehlermeldung *error* ab.
f. Schritt 6: Wenn das Achtfache von OctetLength(n) kleiner als L_N ist,
dann setze diesen Algorithmus mit Schritt 2 fort.
g. Schritt 7: Setze N = Extract_MSBit(n, L_N). <

   c) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

29) Ort (N001.200)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N001.200) K_COS

Tabelle 16 , CosT_7d4: Definition der Funktion RAND(...)

Input:	n	Positive ganze Zahl, welche die Anzahl Oktette in *out* angibt
Output:	*out*	Zufälliger Oktettstring der Länge n Oktett
Errors:	–	Keine
Notation:		*out* = RAND(n)

Das COS MUSS folgende Aktion durchführen: Erzeuge einen Oktettstring *out* der Länge n, wobei jedes Bit von *out* zufällig erzeugt wird. <=

   c) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

30) Ort (N001.210)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Der Inputparameter x wird genauer beschrieben.
   c) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N001.210) K_COS

Tabelle 17 , CosT_c0e: Definition der Funktion ceiling(...)

Input:	x	beliebige reelle Zahl, negativ, null oder positiv
Output:	n	kleinste ganze Zahl, die nicht kleiner ist als x
Errors:	–	Keine
Notation:		n = ceiling(x)

Das COS MUSS folgende Aktion durchführen: Zu x wird die kleinste ganze Zahl n bestimmt, die nicht kleiner ist als x, das heißt: ceiling(x) = min $\{n \in Z \mid n \geq x\}$.

   d) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

31) Ort (N001.220)
   a) Die Tabelle wird in die Anforderung integriert.
   b) Der Inputparameter x wird genauer beschrieben.
   c) Die Anforderungstexte werden um einen Passus mit RFC-Wort ergänzt.

(N001.220) K_COS

Tabelle 18 , CosT_6ed: Definition der Funktion floor(...)

Input:	x	beliebige reelle Zahl, negativ, null oder positiv
Output:	n	größte ganze Zahl, die nicht größer ist als x
Errors:	–	Keine
Notation:		n = floor(x)

Das COS MUSS folgende Aktion durchführen:Zu x wird die größte ganze Zahl n bestimmt, die nicht größer ist als x, das heißt: floor(x) = max $\{n \in Z \mid n \leq x\}$.

   d) Begründung: Der Anforderungstext ist ohne Tabelle schlecht verständlich.

32) Ort (N002.100), (N002.200), (N002.300)
   a) Der Text in (N002.100) wird so ergänzt, dass (N002.200) und (N002.300) überflüssig werden.

(N002.100) K_COS
Das COS MUSS mindestens RSA–Schlüssel mit
a. einer Moduluslänge *modulusLength* aus den Mengen
   1. {2048} Bit für private und öffentliche RSA-Schlüssel unterstützen und
   2. {3072} Bit, falls es sich um einen privaten RSA-Schlüssel handelt und
b. öffentlichen Exponenten e aus folgendem Intervall unterstützen:
   $[2^{16}+1, 2^{32}-1]$ = [65.537, 4.294.967.295] = ['0001 0001', 'FFFF FFFF']. <=

(N002.200) Diese Anforderung ist absichtlich leer. Der ursprüngliche Text erlaubte auch andere Moduluslängen zu unterstützen. Dies wird nun in (N002.100) durch das Wort "mindestens" ausgedrückt.

(N002.300) Diese Anforderung ist absichtlich leer. Der ursprüngliche Text erlaubte auch andere Werte für öffentliche Exponenten. Dies wird nun in (N002.100) durch das Wort "mindestens" ausgedrückt.

   b) Begründung: Vereinfachung der Anforderungslage.

Ab Kapitel 6 werden editorische Änderungen weniger aufwendig dokumentiert.

33) Die Tabelle mit der Signatur der Funktion und deren Parameter wird in die Anforderung integriert und / oder der Anforderungstext wird um einen Passus mit RFC-Wort ergänzt. Dies betrifft folgende Anforderungen:
(N001.290), (N001.300), (N001.310), (N001.320), (N001.500), (N001.510), (N001.520), (N001.530), (N002.000), (N002.010), (N002.500), (N002.810), (N002.820), (N003.010), (N003.020), (N003.200), (N003.300), (N003.400), (N003.500), (N003.600), (N003.800), (N004.000), (N004.200), (N004.400), (N004.500), (N004.700), (N00.4800), (N020.700), (N026.510)

34) Ort (N004.400)

a) Unterpunkt i wird geändert

i. Schritt 9: Setze *maskedDB* = *DB* XOR *dbMasked*.

b) Begründung: Korrektur

35) Die an die COS Hersteller gerichtete Erlaubnis bestimmte Dinge zusätzlich zu implementieren oder eben auch nicht über den geforderten Umfang hinaus Funktionalität anzubieten wird editoriell umgeformt. Dabei wird aus einer "KANN" Anforderung, die sich an das COS richtet, eine "MUSS" Anforderung für den funktionalen Test. Beispiel:

(N002.609) K_TST

~~Das COS KANN~~ Für die funktionale Eignung MUSS es zulässig sein, dass das COS weitere ellipitsche Kurven

a. unterstützt oder

b. ablehnt.

Dies betrifft folgende Anforderungen:
(N002.609), (N007.100)b, (N007.170)d, (N007.170)e, (N007.200)b, (N007.600)b, (N007.700)b, (N007.900)b, (N008.000)b, (N008.120)b, (N009.600)b, (N009.860), (N010.000)b, (N010.010), (N010.400)b, (N011.060), (N011.200)b, (N011.200)c, (N011.300)b, (N011.300)c, (N011.500)b, (N012.000), (N012.200)b, (N012.400)b, (N012.500)b, (N012.800), (N013.000)b, (N014.800), (N015.000)b, (N015.300)b, (N015.310)b, (N015.400)b, (N015.500)b, (N015.800)b, (N015.800)d, (N016.000)b, (N016.100)b, (N016.300), (N016.335), (N016.400)b, (N016.590)b, (N017.000), (N017.020)b, (N017.030)c, (N017.032)b, (N017.100)b, (N017.400)d, (N017.600)b, (N017.620)b, (N017.900)b, (N018.300), (N018.422), (N018.822), (N019.100)b, (N019.200)b, (N019.500)b, (N019.600)b, (N019.822)b, (N019.824,)b, (N019.900)b.4, (N019.900)e.5, (N019.900)g.4, (N019.920)b, (N019.920)c, (N020.400), (N021.638)d, (N021.800)d, (N023.740)b, (N023.820)d, (N023.820)e, (N024.100)e.2, (N024.810)b, (N024.820)b, (N025.310)b.2, (N025.310)c.3, (N026.900)b, (N029.878)b, (N029.880)b, (N029.890)c, (N029.900)e.2, (N029.900)f.2, (N029.900)h.2, (N029.900)i.2, (N029.900)j,2, (N029.900)l.2, (N031.700)a.1.iii, (N031.940), (N034.400), (N034.500), (N034.900), (N035.000)b, (N035.500), (N035.900), (N036.100), (N036.200)b, (N036.700), (N037.200), (N037.300)b, (N037.800)b.2, (N037.850), (N037.900), (N038.900), (N039.000)b, (N039.010), (N039.300), (N039.700), (N047.400), (N047.500)b, (N047.600), (N048.000)a.3, (N048.738), (N048.740)b, (N048.760), (N048.838), (N048.840)b, (N048.948), (N048.950)b, (N048.966), (N049.500), (N049.600)b, (N050.100)b, (N050.300), (N051.600), (N051.700)b, (N052.938), (N052.940)b, (N052.950)b, (N052.956), (N053.700), (N053.800)b, (N054.400)b, (N054.600), (N055.240), (N055.224)b, (N055.254)b, (N055.260), (N056.700), (N056.800)b, (N057.400)b, (N057.900), (N058.800), (N058.900)b, (N059.650)b.2, (N059.900), (N061.900), (N062.000)b, (N062.600)b, (N062.900), (N063.428), (N063.430)b, (N063.448), (N064.000), (N064.100)b, (N064.700)b, (N064.900), (N066.200), (N066.300)b, (N068.500), (N068.600)b, (N070.800), (N070.900)b, (N071.750)b.2, (N071.900), (N072.500), (N072.600), (N072.700), (N073.800), (N074.100), (N075.300)b, (N075.300)b, (N075.600), (N075.700)b, (N076.120)b, (N078.600), (N076.900)b, (N077.320)b, (N078.000), (N078.10)b, (N080.400), (N080.500)b, (N082.300), (N082.400)b, (N084.280), (N084.048), (N085.050)d, (N085.500), (N085.600)b, (N086.500), (N086.600)b, (N087.232), (N087.236)b, (N088.100), (N088.200)b, (N089.900), (N090.000)c, (N091.500), (N091.600)d, (N092.200), (N093.700), (N093.800)b, (N095.600), (N095.700)b, (N096.000), (N096.350), (N096.360)b, (N096.390), (N096.392)b, (N096.400), (N096.470)b, (N097.300), (N097.400)b, (N097.500)a.1, (N097.900), (N098.700), (N098.800)b, (N098.900), (N099.340), (N099.344)b, (N099.400), (N099.458), (N099.460)b, (N099.461), (N099.462)c, (N099.540), (N099.542)c, (N099.545), (N100.300)b, (N100.800)b, (N101.300)b, (N101.800)b, (N102.300)b, (N102.800)b, (N103.845)b, (N103.900), (N104.000)e, (N104.100), (N104.302)a.2, (N104.700), (N104.900), (N106.500), (N252.100)b, (N252.200)b, (N252.300)a, (N252.300)c.2, (N253.140)

36) Ort (N004.490)

a) (N004.490) wird aufgeteilt in (N004.490) und (N004.492)

6.8.1.3 Elliptic Curve Key Agreement

Die hier beschriebene Funktionalität ist an der physikalischen Schnittstelle nicht direkt sichtbar, wird aber im Rahmen verschiedener Funktionen mit elliptischen Kurven verwendet.

Hinweis CosH_c0c: In einer früheren Dokumentenversion wurde in diesem Kapitel die Funktion ECKA(...) spezifiziert. In dieser Dokumentenversion wurde die Funktion ECKA(...) aufgespalten in ECKApoint(...) und ECKAvalue(...).

6.8.1.3.1 Elliptic Curve Key Agreement Point Sab

Die hier beschriebene Funktionalität berechnet aus einem privaten und einem öffentlichen Schlüssel auf einer elliptischen Kurve einen Punkt, der als gemeinsames Geheimnis weiterverwendet wird.

Hinweis CosH_261: Dies entspricht dem ersten Teil der Funktion ECKA(...) aus früheren Dokumentenversionen. Der übrige Teil der Funktion ECKA(...) ist in (N004.492) enthalten.

(N004.490) K_COS

Tabelle 54 , CosT_eb0: Definition der Funktion ECKApoint(...)

Input:	d	natürliche Zahl, die dem privaten Schlüssel PrK entspricht
	P	Punkt auf derselben elliptischen Kurve, wie der private Schlüssel PrK
	dP	Domainparameter gemäß (N008.600)
Output:	S_{AB}	"gemeinsamer geheimer Punkt"
Errors:	ERROR	Wenn das Ergebnis in Schritt 3 der unendlich ferne Punkt ist
Notation:		S_{AB} = ECKApoint(d, P, dP)

Das COS MUSS S_{AB} mittels d, P und dP berechnen, wobei die Schritte 1 bis 3 [BSI-TR-03111#4.3.1] entsprechen. Es gelten folgende Definitionen:
$h = dP.h$, $n = dP.n$, $L = dP.L$
a. Schritt 1: $l = h^{-1} \bmod n$.
b. Schritt 2: $Q = [\, h \,] P$.
c. Schritt 3: $S_{AB} = [\, d\, l \bmod n \,] Q$.
Wenn S_{AB} gleich dem unendlich fernen Punkt O der Kurve ist,
dann gebe den Fehler "ERROR" zurück und beende diesen Algorithmus,
sonst gebe S_{AB} zurück.
d. Dieser Punkt ist absichtlich leer. Er wurde nach (N004.492) verschoben. ~~Schritt 4: K_{AB} = I2OS(x_S, L).~~ <=

6.8.1.3.2 Elliptic Curve Key Agreement Value Zab

Die hier beschriebene Funktionalität wandelt einen Punkt auf einer elliptischen Kurve, der ein gemeinsames Geheimnis darstellt, in einen Oktettstring um.

Hinweis CosH_35e: Dies entspricht dem zweitenTeil der Funktion ECKA(...) aus früheren Dokumentenversionen. Der erste Teil der Funktion ECKA(...) ist in (N004.490) enthalten.

6.8.1.3.2 Elliptic Curve Key Agreement Value Zab

Die hier beschriebene Funktionalität wandelt einen Punkt auf einer elliptischen Kurve, der ein gemeinsames Geheimnis darstellt, in einen Oktettstring um.

Hinweis CosH_35e: Dies entspricht dem zweitenTeil der Funktion ECKA(...) aus früheren Dokumentenversionen. Der erste Teil der Funktion ECKA(...) ist in (N004.490) enthalten.

(N004.492) K_COS

Tabelle 55 , CosT_e3a: Definition der Funktion ECKAvalue(...)

Input:	d	natürliche Zahl, die dem privaten Schlüssel PrK entspricht
	P	Punkt auf derselben elliptischen Kurve, wie der private Schlüssel PrK
	dP	Domainparameter gemäß (N008.600)
Output:	Z_{AB}	"gemeinsamer geheimer Oktettstring"
Errors:	$ERROR$	Wenn die Funktion in Schritt 1 diesen Fehler meldet
Notation:		Z_{AB} = ECKAvalue(d, P, dP)

Das COS MUSS Z_{AB} mittels d, P und dP berechnen, wobei dies Schritt 4 in [BSI-TR-03111#4.3.1] entspricht.
a. Schritt 4.a: S_{AB} = ECKApoint(d, P, dP).
Wenn der Aufruf der Funktion ECKApoint(...) mit der Fehlermeldung $ERROR$ abbricht,
dann gebe $ERROR$ zurück und beende diesen Algorithmus,
sonst fahre mit Schritt 4.b fort.
b. Schritt 4.b: Z_{AB} = I2OS(x_s, $dp.L$) mit x_s == x-Komponente des Punktes S_{AB}. <=

b) Begründung: Fehlerbeseitigung in (N085.064)b.4 und (N085.066)c.2

37) Ort (N076.500)

a) (N076.500) wird in drei Anforderungen aufgeteilt und (N076.500)b wird editorisch umgeformt.

(N076.500)a K_COS
Wenn *affectedObject.flagEnabled* den Wert False besitzt, genau dann MUSS als Trailer NoError verwendet werden.

(N076.500)b K_COS
Wenn *affectedObject.flagEnabled* den Wert True besitzt, dann ~~ist~~ MUSS PasswordBlocked eine höhere Priorität als PasswordNotUsable haben. Die Priorität der übrigen Trailer in CosT_255 ist ~~bis auf folgende Ausnahme~~ herstellerspezifisch.
~~1. PasswordBlocked MUSS eine höhere Priorität als PasswordNotUsable haben.~~

(N076.500)c K_COS
Jeder Trailer in CosT_255 MUSS eine höhere Priorität als WrongSecretWarning haben.

b) Begründung: Die Aufteilung erleichtert die Zuweisung von Testfällen. Aus der Umformung wird deutlicher, dass es sich um eine Anforderung an K_COS handelt.

38) Ort (N077.700)

a) (N077.700) wird in drei Polarion-Anforderungen aufgeteilt und (N077.700)b wird editorisch umgeformt.

(N077.700)a K_COS
Wenn *affectedObject.flagEnabled* den Wert True besitzt, genau dann MUSS als Trailer NoError verwendet werden.

(N077.700)b K_COS
Wenn *affectedObject.flagEnabled* den Wert False besitzt, dann ~~ist~~ MUSS PasswordBlocked eine höhere Priorität als PasswordNotUsable haben. Die Priorität der übrigen Trailer in CosT_903 ist ~~bis auf folgende Ausnahme~~ herstellerspezifisch.
~~1. PasswordBlocked MUSS eine höhere Priorität als PasswordNotUsable haben.~~

(N077.700)c K_COS
Jeder Trailer in CosT_903 MUSS eine höhere Priorität als WrongSecretWarning haben.

b) Begründung: Die Aufteilung erleichtert die Zuweisung von Testfällen. Aus der Umformung wird deutlicher, dass es sich um eine Anforderung an K_COS handelt.

39) Ort (N083.300)

a) (N083.300) wird in drei Polarion-Anforderungen aufgeteilt und (N083.300)a wird editorisch umgeformt.

(N083.300)a K_COS
PasswordBlocked MUSS eine höhere Priorität als PasswordNotUsable haben. Die Priorität der übrigen Trailer in CosT_49d ist ~~bis auf folgende Ausnahme~~ herstellerspezifisch.
~~1. PasswordBlocked MUSS eine höhere Priorität als PasswordNotUsable haben.~~ <=

(N083.300).b K_COS
Jeder Trailer in CosT_49d MUSS eine höhere Priorität als WrongSecretWarning haben. <=

(N083.300)c) K_COS
WrongSecretWarning MUSS eine höhere Priorität als NoError haben. <=

b) Begründung: Die Aufteilung erleichtert die Zuweisung von Testfällen. Aus der Umformung wird deutlicher, dass es sich um eine Anforderung an K_COS handelt.

40) Ort (N099.356)b

a) (N099.356) wird in zwei Polarion-Anforderungen aufgeteilt und (N099.356)b wird geändert.

(N099.356)a K_COS
Es MUSS *random* = RAND(Ne) gesetzt werden. <=

(N099.356)b K_PP-COS
Die Güte der Zufallszahl in *random* MUSS im zugehörigen Schutzprofil festgelegt werden.~~mindestens [BSI-TR-03116-1#3.5] PTG.3 oder K4-DRNG oder DRG.3 entsprechen.~~ <=

b) Begründung: Die Aufteilung ist notwendig, da sich a und b an unterschiedliche Entitäten richten. Die Änderung in b ist notwendig, da das Schutzprofil Anforderungen an die Güte erhebt, die sich nicht nur aus der TR-03116-1 ableiten und die über die aktuellen Anforderungen in gemSpec_COS hinausgehen.

41) Ort: (N008.600)
   a) Referenzierende Stellen werden präzisiert, zudem wurde τ durch t ersetzt

(N008.600) K_Anwendungsspezifikation {K_Karte}
Der Attributstyp *domainParameter* MUSS enthalten

a. alle Parameter aus ~~Tabelle 7~~ [BSI-TR-03111#Tabelle 2.1] und
b. eine Zahl L, welche die minimale Anzahl Oktette angibt, die nötig sind, um p als vorzeichenlose Zahl zu codieren. Dieser Parameter wurde ~~der "offiziellen" Liste der Domainparameter~~ [BSI-TR-03111#Tabelle 2.1] hinzugefügt, da dieser Parameter in diesem Dokument vielfach verwendet wird. Allgemein gilt L=ceiling($\log_{256} p$). Wegen (N002.500) gilt:
   1. L = 32 für brainpoolP256r1 und ansix9p256r1.
   2. L = 48 für brainpoolP384r1 und ansix9p384r1.
   3. L = 64 für brainpoolP512r1
c. eine Zahl t, welche die Bitlänge von n angibt (siehe [BSI-TR-03111#Table 1.1], dort mit dem griechischen Buchstaben tau bezeichnet). Dieser Parameter wurde ~~der "offiziellen" Liste der Domainparameter~~ [BSI-TR-03111#Tabelle 2.1] hinzugefügt, da dieser Parameter in diesem Dokument vielfach verwendet wird. Allgemein gilt t=ceiling($\log_2 n$). Wegen (N002.500) gilt:
   1. t = 256 für brainpoolP256r1 und ansix9p256r1.
   2. t = 384 für brainpoolP384r1 und ansix9p384r1.
   3. t = 512 für brainpoolP512r1.
d. einen Oktettstring *OID*, durch den die Domainparameter (p, a, b, G, n, h) weltweit eindeutig bestimmt werden. Dieser Parameter wurde ~~der "offiziellen" Liste der Domainparameter~~ [BSI-TR-03111#Tabelle 2.1] hinzugefügt, da in diesem Dokument Domainparameter vielfach per *OID* referenziert werden. *OID* MUSS so aus CosT_a91 gewählt werden, dass damit eine elliptische Kurve gemäß (N002.500) referenziert wird.

   b) Begründung: Präzisierung, τ ist in Polarion nicht darstellbar.

42) Ort (N010.000)a.1
   a) Referenz geändert

(N010.000)a.1 K_Anwendungsspezifikation {K_Karte}
Unter Berücksichtigung der maximalen Schachtelungstiefe (siehe (N020.390)~~(N020.400)~~) MÜSSEN

   b) Begründung: Korrektur

43) Ort (N019.930)
   a) Änderung des Anforderungstextes

(N019.930) K_Anwendungsspezifikation {K_Karte}
Die Werte aller Attribute *applicationIdentifier* MÜSSEN paarweise verschieden sein. ~~Im Objektsystem DARF es KEINE zwei verschiedenen Ordner geben, deren Attribut *applicationIdentifier* identisch ist.~~

   b) Begründung: Präzisierung

44) Ort Einleitung des Kapitels 9.2.3.1
   a) Der Text wird ergänzt

9.2.3.1 Suche nach einem geheimen Schlüsselobjekt

In diesem Kapitel werden symmetrische Authentisierungsobjekte gemäß CosK_5fa, symmetrische Kartenverbindungsobjekte gemäß CosK_24a und private Schlüsselobjekte gemäß CosK_ab8 gemeinsam behandelt, da nach ihnen auf gleiche Art und Weise gesucht wird. Sie werden im Rahmen diverser kryptographischer Operationen verwendet. Symmetrische Authentisierungsobjekte und symmetrische Kartenverbindungsobjekte werden zudem auch bei der Auswertung von Zugriffsregeln (siehe (N022.300)) verwendet. Bei dieser Art der Schlüsselsuche handelt es sich um eine karteninterne Funktionalität, bei der viele Implementierungsdetails eine Rolle spielen. An der Schnittstelle "Interpreter" (siehe CosA_e09) wird das im Folgenden festgelegte Verhalten sichtbar:

   b) Begründung: Vervollständigung

45) Ort (N021.800)a, b, c
   a) Der Anforderungstext wird ergänzt

(N021.800)a K_Anwendungsspezifikation {K_Karte}
Ein Listenelement MUSS den Namen eines Kommandos enthalten, der gemäß CosK_3a7 äquivalent ist zu genau einer Kombination aus CLA Byte (siehe CosK_90e) und INS Byte (siehe CosK_19e). <=

(N021.800)a.1 K_Anwendungsspezifikation {K_Karte}
Das CLA Byte in einer Zugriffsart MUSS die Kanalnummer null enthalten. Das ist so zu interpretieren, dass die Zugriffsart für beliebige Kanalnummern gilt. <=

(N021.800)a.2 K_Anwendungsspezifikation {K_Karte}
Das CLA Byte in einer Zugriffsart DARF KEIN Secure Messaging anzeigen. Das ist so zu interpretieren, dass die Zugriffsart sowohl für ungeschützte Nachrichtenübertragung, als auch für beliebig per Secure Messaging geschützte Nachrichtenübertragung gilt. <=

Innerhalb eines Listenelementes ist der Parameter P1 (siehe CosK_66b) optional. Das bedeutet:

(N021.800)b.1 K_Anwendungsspezifikation {K_Karte}
Es MUSS möglich sein, dass ein Listenelement genau einen Wert für den Parameter P1 enthält. Das ist so zu interpretieren, dass die Zugriffsart nur für diesen Wert von P1 gilt. <=

(N021.800)b.2 K_Anwendungsspezifikation {K_Karte}
Es MUSS möglich sein, dass ein Listenelement keinen Parameter P1 enthält. Das ist so zu interpretieren, dass die Zugriffsart für beliebige Werte von P1 gilt. <=

Innerhalb eines Listenelementes ist der Parameter P2 (siehe CosK_165) optional. Das bedeutet:

(N021.800)c.1 K_Anwendungsspezifikation {K_Karte}
Es MUSS möglich sein, dass ein Listenelement genau einen Wert für den Parameter P2 enthält. Das ist so zu interpretieren, dass die Zugriffsart nur für diesen Wert von P2 gilt. <=

(N021.800)c.2 K_Anwendungsspezifikation {K_Karte}
Es MUSS möglich sein, dass ein Listenelement keinen Parameter P2 enthält. Das ist so zu interpretieren, dass die Zugriffsart für beliebige Werte von P2 gilt. <=

   b) Begründung: Präzisierung.

46) Ort (N022.210)
   a) Eine neue Anforderung ersetzt (N022.200)a.1.i und (N022.200)a.1.ii.

(N022.210) K_TST
Wenn durch das Dekrementieren gemäß (N022.200)a.1 der Wert auf Null gefallen ist, dann MUSS es für die funktionale Eignung zulässig sein, dass das COS den Eintrag für dieses Passwort in einer der Listen *globalPasswordList* oder *dfSpecificPasswordList*
a. verbleibt, oder
b. entfernt.

   b) Begründung: Anderer Adressat des ursprünglichen Anforderungstextes.

47) Dieser Punkt ist absichtlich leer.

48) Ort: Einleitung zu Kapitel 11.6.1
   a) Der Text wird so geändert, dass er konsistent zu den Anforderungen des Kapitels wird.

11.6.1 CosK_3c7 Datenfeld

Das Datenfeld *rspData* einer Antwort-APDU ist optional in dem Sinne, dass es möglicherweise leer ist. ~~Es ist möglich, dass es fehlt.~~ Das Datenfeld *rspData* ist ein Oktettstring der Länge Nr. Gemäß [ISO/IEC 7816-3] ist der Definitionsbereich von Nr {0, 1, ..., 65536}.

   b) Begründung: Konsistenz

49) Ort (N031.700)a.1
   a) Der Text wird in drei Anforderungen aufgeteilt:

(N031.700)a.1.i K_externeWelt {K_Karte}
Die externe Welt MUSS die gesicherte Kommanod-APDU CmdApdu2 gemäß den Vorgaben aus CosK_842 codieren. <=

(N031.700)a.1.ii K_COS
Schritt 1: Wenn ein DO mit Tag = '87' in der CmdApdu2 vorhanden ist, dann MUSS *flagCmdEnc* auf True gesetzt werden, sonst auf False. <=

(N031.700)a.1.iii K_TST
~~Das COS KANN weitere~~ Für die funktionale Eignung MUSS es zulässig sein, dass das COS weitere Secure Messaging Formate
A. unterstützt oder
B. abgelehnt. <=

   b) Begründung: Unterschiedliche RFC-Worte, unterschiedliche Adressaten.

50) Ort Einleitung Kapitel 14.3
   a) Der Einleitungstext wird ergänzt:

14.3 Zugriff auf Daten in transparenten EF

Es ist möglich auf Daten im *body* eines transparenten EF lesend (READ BINARY-Kommando) oder schreibend (UPDATE BINARY-Kommando) zuzugreifen. Zudem ist es möglich den Dateninhalt von *body* durch nullen zu löschen (ERASE BINARY-Kommando). Sofern die maximale Dateigröße nicht erreicht ist, ist es möglich *body* um weitere Daten zu ergänzen (WRITE BINARY-Kommando). Die Größe des Teils von *body*, der für Lesezugriffe zugänglich ist, lässt sich auch verringern (SET LOGIGAL EOF-Kommando).

   b) Begründung: Vervollständigung der Use Cases.

51) Ort List der Use Cases Eingangs von 14.6.4 GET PIN STATUS
   a) Ein weiterer Punkt wird ergänzt:

Das Kommando GET PIN STATUS zeigt in den Antwortdaten an,

- ob der Sicherheitszustand des Passwortobjektes gesetzt ist.
- welchen Wert das Attribut *retryCounter* besitzt.
- ob ein Passwortobjekt mit einem Transportschutz versehen ist und falls ja, welches Transportschutzverfahren vom Passwortobjekt verwendet wird.
- ob eine Benutzerverifikation zum Setzen des Sicherheitsstatus erforderlich ist.

   b) Begründung: Vervollständigung der Liste.

52) Ort Kommando GET PIN STATUS:
   a) Folgende Änderungen:

i) Der Trailer '63 C0' wird aus der Tabelle mit Fehlercodes in die Tabelle mit Trailern im Erfolgsfall verschoben.

Tabelle 217, CosT_a97: GET PIN STATUS Antwort-APDU im Erfolgsfall

Trailer	Inhalt	Beschreibung
'62 Cx': '62 C1' '62 C7'	TransportStatus Transport–PIN Leer–PIN	Passwortobjekt eingeschaltet, Passwortobjekt ist mit Transportschutz versehen, Transportschutzverfahren enthalten im Least Significant Nibble (siehe auch CosK_6b8)
'62 D0'	PasswordDisabled	Passwortobjekt ausgeschaltet, Verifikation nicht erforderlich
'63 Cx'	RetryCounter, x ungleich null	Passwortobjekt eingeschaltet, Passwortobjekt ohne Transportschutz (das bedeutet regularPassword), Wert des Fehlbedienungszählers enthalten im Least Significant Nibble
'63 C0'	RetryCounter, x gleich null	Passwortobjekt eingeschaltet, Passwortobjekt wegen Wert des Fehlbedienungszählers blockiert Blockade ist möglicherweise mittels RESET RETRY COUNTER aufhebbar
'90 00'	NoError	Passwortobjekt eingeschaltet, Sicherheitszustand gesetzt

Tabelle 218, CosT_973: GET PIN STATUS Antwort-APDU im Fehlerfall

Trailer	Inhalt	Beschreibung
~~'63 C0'~~	~~'63 Cx' = RetryCounter, x gleich null~~	~~Passwortobjekt eingeschaltet, Passwortobjekt wegen Wert des Fehlbedienungszählers blockiert~~
'69 82'	SecurityStatusNotSatisfied	Zugriffsregel nicht erfüllt
'6A 88'	PasswordNotFound	Adressiertes Passwort wurde nicht gefunden

ii) (N078.800) wird geändert

(N078.800)a K_TST
Die Priorität der Trailer in CosT_973 MUSS herstellerspezifisch sein.

(N078.800)b.1 K_COS
Jeder Trailer in CosT_973 MUSS eine höhere Priorität als RetryCounter = '63 C0" haben.

(N078.800)b.2 K_COS
RetryCounter = '63 C0' MUSS eine höhere Priorität als PasswordDisabled haben.

(N078.800)c K_COS
PasswordDisabled MUSS eine höhere Priorität als NoError haben.

(N078.800)d K_COS
NoError MUSS eine höhere Priorität als TransportStatus haben.

(N078.800)e K_COS
TransportStatus MUSS eine höhere Priorität als RetryCounter = '63 Cx' mit x ungleich null haben.

b) Begründungen:
i) '63 xx' ist eine Warnung, also eine erfolgreiches Kommando, kein Fehlerfall.
ii) Die geänderte Fassung gibt deutlicher das gewünschte Verhalten an.

53) Ort (N081.800)b
a) Die Afo wird in zwei Teile gespalten

(N081.800)b.1 K_TST
Für diie funktionale Eignung MUSS es zulässig sein, dass die Änderungen von *secret*, *transportStatus* oder *retryCounter* in eigenen Transaktionen stattfinden. <=

(N081.800)b.2 K_COS
Wenn die Änderungen von *secret*, *transportStatus* oder *retryCounter* in eigenen Transaktionen stattfinden, dann MUSS *retryCounter* als letztes geändert werden. <=

b) Begründung: Unterschiedliche Adressaten.

54) Ort Kapitel 14.7.1
a) Hinweis 107 wird ersatzlos entfernt. Als Seiteneffekt werden [gemSpec_eGK_P1] und [HBA_P1] aus der Liste der referenzierten Dokumente entfernt.

~~*Hinweis (107): In [gemSpec_eGK_P1] wurden die Varianten EXTERNAL versus MUTUAL anhand eines Algorithm Identifiers unterschieden. Hier werden die Varianten, wie schon in [HBA_P1], anhand des LeFeldes unterschieden*~~

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
~~[gemSpec_eGK_P1]~~	~~gematik: Teil 1 – Spezifikation der elektrischen Schnittstelle (Vorgängerversion dieses Dokumentes für Karten der Generation 1)~~
~~[HBA_P1]~~	~~Spezifikation des elektronischen Heilberufsausweises Teil I: Kommandos, Algorithmen und Funktionen der Betriebssystemplattform, Version 2.3.2, 05.08.2009 (Vorgängerversion dieses Dokumentes für Karten der Generation 1)~~

b) Begründung: Mangelnde Relevanz.

55) Ort (N085.064)b.4 und (N085.066)c.2

a) Die Berechnung von ~G wird geändert:

(N085.064)b.3-6 K_COS, Option_kontaktlose_Schnittstelle

b.3: Es wird ein ephemeres Schlüsselpaar generiert:

i. $\sim SK1_{PICC}$ = zufällige Zahl aus dem Intervall [1, $dP.n - 1$],

ii. $\sim PK1_{PICC} = \sim SK1_{PICC} * dP.G$,

b.4: Berechne den ephemeren Basispunkt $\sim G$ mit s aus Schritt 1:

$\sim G = s * dP.G$ + ECKApoint($\sim SK1_{PICC}$, $\sim PK1_{PCD}$, dP) ~~$\sim SK1_{PICC} * \sim PK1_{PCD}$~~ ,

b)

(N085.066)c.2-5 K_COS, Option_PACE_PCD

c.2: Berechne den ephemeren Basispunkt $\sim G$ mit s aus Schritt 2:

$\sim G = s * dP.G$ + ECKApoint($\sim SK1_{PCD}$, $\sim PK1_{PICC}$, dP) ~~$\sim SK1_{PCD} * \sim PK1_{PICC}$~~ ,

c) Begründung: In einer früheren Dokumentenversion war der zweite Summand in (N085.064)b.4 definiert als Produkt aus $\sim SK1_{PICC}$ und $\sim PK1_{PCD}$. Dies ist nicht konform zu [BSI-TR-03110-3#A.3.4.1]. Deshalb wurde (N085.064)b.4 konform zu [BSI-TR-03110-3#A.3.4.1] korrigiert. Für die normativ zu unterstützenden elliptischen Kurven aus (N002.500) hat diese Änderung keine Auswirkungen, da für alle diese Kurven gilt: Cofaktor h = 1

56) Ort (N085.066)d.4.iii

a) Die OID wird geändert

(N085.066)d.2-7 K_COS, Option_PACE_PCD

d.2: Berechne gemeinsames Geheimnis zur Ableitung von Sessionkeys:

i. KD.i = K_{AB} = ECKAvalue($\sim SK2_{PCD}$, $\sim PK2_{PICC}$, $\sim D$),

ii. KD.e = I2OS(0, OctetLength(KD.i)) = '00…00',

d.3: Die *algID* aus *channelContext.keyReferenceList.internalAuthenticate* bestimmt, wie der Schlüssel k_{MAC} berechnet wird: k_{MAC} = KDF(KD.i, 2, *algID*),

d.4: Für den Zusammenhang zwischen *algID* aus *externalAuthenticate* oder *internalAuthenticate* in *channelContext.keyReferenceList* und *OID* gilt hier:

i. *algID* = id-PACE-PCD-ECDH-GM-AES-CBC-CMAC-128 => *OID* = id-PACE-ECDH-GM-AES-CBC-CMAC-128

ii. *algID* = id-PACE-PCD-ECDH-GM-AES-CBC-CMAC-192 => *OID* = id-PACE-ECDH-GM-AES-CBC-CMAC-192

iii. *algID* = id-PACE-PCD-ECDH-GM-AES-CBC-CMAC-256 => *OID* = id-PACE-ECDH-GM-AES-CBC-CMAC-256 ~~128~~

b) Begründung: Korrektur eines Copy & Paste Fehlers.

57) Ort Kapitel 14.8

a) Überschrift geändert

14.8 CosK_00c Kryptographische Operationen

Hinweis CosH_b01: In einer früheren Version trug dieses Kapitel die Überschrift "Kryptoboxkommandos". Es wurde umbenannt in "Kryptographische Operationen", um Verwechselungen mit der Option_Kryptobox zu vermeiden.

b) Begründung: Vermeidung von Verwechselungen

58) Ort (N096.381)

a) Verweise werden geändert:

(N096.381) K_externeWelt {K_Karte}

Der Parameter PO_B enthält einen Punkt auf einer elliptischen Kurve. Dieser Punkt repräsentiert den öffentlichen Schlüssel des Signierenden. Der Parameter PO_B ist ein Oktettstring, dessen Inhalt so gewählt werden SOLL, dass bei der Decodierung kein Fehler auftritt (siehe (N096.396)a Funktion O2P(…) und (N000.300)~~(N004.500)a~~).

b) Begründung: Korrektur.

59) Ort (N097.700)a
a) Der Text wird ergänzt

(N097.700) K_COS
Wenn *operationMode* Element der Menge {'80', '84', 'C0', 'C4'} ist, dann MÜSSEN folgende Schritte ausgeführt werden:

a. Es MUSS ein Schlüsselpaar (*PrK*, *PuK*) erzeugt werden, dessen Eigenschaften zu den Attributen von *affectedObject* und zu (N002.100) bzw. (N002.500) passen.
b. Anschließend MUSS das Attribute *keyAvailable* mit Transaktionsschutz auf den Wert True geändert werden.

b) Begründung: Nennung von ELC-Schlüsseln neben der Nennung von RSA Schlüsseln.

60) Ort (N099.356)b
a) Adressat und Text werden geändert

(N099.356)b K_PP-COS Blattanforderung: ML-79683
Die Güte der Zufallszahl in *random* MUSS im zugehörigen Schutzprofil festgelegt werden ~~mindestens [BSI-TR-03116-1#3.5] PTG.3 oder K4-DRNG oder DRG.3 entsprechen.~~ <=

b) Begründung: Anpassung an die Wirklichkeit. Auch heute schon gehen die Anforderungen im PP-COS über die Anforderungen in BSI-TR-03116 hinaus.

61) Ort (N099.600)c
a) Der Text wird geändert

(N099.600) K_externeWelt {K_Karte}
Falls ein symmetrisches Authentisierungsobjekt oder ein symmetrisches Kartenverbindunsobjekt oder ein privates Schlüsselobjekt referenziert wird, dann besteht der Parameter *keyReference* aus den zwei Teilen *location* und *identifier*. Der Teil *location* zeigt an, ob ein globaler oder DF-spezifischer Schlüssel von der Aktion betroffen ist. Als Wert für *location* MUSS ein Element der Menge {'00', '80'} verwendet werden. Dabei gilt:

a. Der Wert *location* = '00' MUSS verwendet werden, wenn ein globaler Schlüssel betroffen ist.
b. Der Wert *location* = '80' MUSS verwendet werden, wenn ein DF-spezifischer Schlüssel betroffen ist.
c. Der Parameter *identifier* bestimmt das betroffene Schlüsselobjekt. Der Wert von *identifier* MUSS konform zu (N016.400)a, (N017.020)a oder (N017.100)a gewählt werden.
d. Der Parameter *keyReference* MUSS in einem Oktett mit folgendem Wert codiert werden:
keyReference = *location* + *identifier*.

b) Begründung: Ergänzung, Präzisierung von Referenzen

62) Ort (N106.200)
a) Der Text wird geändert:

(N106.200) K_externeWelt {K_Karte}
Das zweite Kommando der Sequenz MUSS über Channel_a zum COSa gesendet werden und MUSS ein INTERNAL AUTHENTICATE gemäß (N086.400) mit *algId* gleich symSessionkey4TC sein. Dabei MUSS *cmdData* ein *token* (siehe (N086.902)a.4) mit R1 und *iccsn8* des COSb gemäß (N084.410)a.6.ii enthalten . Die Antwortdaten des COSa werden mit rspData.2 bezeichnet.

b) Begründung: Korrektur und Präzisierung

63) Ort Kapitel 15.4.4, Punkt 8
   a) Im nicht-normativen Text wird die Bezeichnung der Funktion an die in 36) beschriebene Änderung angepasst

8. Im zweiten Schritt werden die öffentlichen Teile der ephemeren Schlüsselpaare in die jeweils andere Komponente importiert und in beiden Komponenten werden je zwei gemeinsame Geheimnisse berechnet und konkateniert. Im Einzelnen:
   a. In Komponente A wird
      i. der ephemere Schlüssel $PK.b$ importiert
      ii. das erste gemeinsame Geheimnis K1 = ECKAvalue($PrK.a$, $PK.b$, D) berechnet.
      iii. das zweite gemeinsame Geheimnis K2 = ECKAvalue($SK.a$, $PuK.b$, D) berechnet.
      iv. Komponente A bildet die Schlüsselableitungsdaten K = K1 || K2.
      v. Der Sicherheitszustand der Komponente A wird entsprechend den Informationen aus dem CV-Zertifikat zu $PuK.b$ gesetzt.
      vi. Mittels der Schlüsselableitungsdaten K werden Sessionkeys berechnet.
   b. Gleichzeitig wird in der Komponente B
      i. der ephemere Schlüssel $PK.a$ importiert
      ii. das erste gemeinsame Geheimnis K1 = ECKAvalue($SK.b$, $PuK.a$, D) berechnet.
      iii. das zweite gemeinsame Geheimnis K2 = ECKAvalue($PrK.b$, $PK.a$, D) berechnet.
      iv. Komponente B bildet die Schlüsselableitungsdaten K = K1 || K2

   b) Begründung: Konsistenz

64) Ort Glossar:
   a) Aus dem Glossar wird der Begriff "schwacher DES-Schlüssel" entfernt.

~~schwacher DES-Schlüssel~~	~~Ein schwacher oder halbschwacher DES-Schlüssel senkt die Stärke einer Verschlüsselung. Zu weiteren Informationen und einer Liste schwacher und halbschwacher Schlüssel siehe:~~ • ~~Schneier, B., "Applied Cryptography, Protocols, Algorithms, and Source Code in C", 2nd edition.~~ • ~~RFC 2409 Appendix A~~ • ~~http://en.wikipedia.org/wiki/Weak_key~~

   b) Begründung. DES-Schlüssel wurden aus dem normativen Umfang von gemSpec_COS entfernt.

65) Ort Referenzierte Dokumente
   a) Die Versionsangaben zu BSI Dokumenten werden aktualisiert.

[BSI-TR-03110-2]	Technical Guideline TR-03110-2, Advanced Security Mechanisms for Machine Readable Travel Documents – Part 2 – Extended Access Control Version 2 (EACv2), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI), Version 2.20, 2015-02-03
[BSI-TR-03110-3]	Technical Guideline TR-03110-3, Advanced Security Mechanisms for Machine Readable Travel Documents – Part 3 – Common Specifications Version 2.20, 2015-02-03
[BSI-TR-03111]	Technical Guideline TR-03111, Elliptic Curve Cryptography, Version 2.10, 2018-06-01
[BSI-TR-03116-1]	Technische Richtlinie BSI TR-03116-1, Kryptographische Vorgaben für Projekte der Bundesregierung, Version 3.20, Datum 21.09.2018, Status: Veröffentlichung, Fassung Septembber 2018

   b) Begründung: Modernisierung

3 Literaturstellen

[gemSpec_COS]	gematik: Spezifikation des Card Operating System (COS), Elektrische Schnittstelle