1 Einleitung

1.1 Technischer Hintergrund

Diese Anlage beschreibt eine vorgeschlagene Änderung zur Spezifikation [gemSpec_KT].

In diesem Kapitel wird der technische Hintergrund der Änderung soweit beschrieben, wie es für das Verständnis des Interoperabilitätsproblems erforderlich ist.

1.1.1 Lokale PIN-Eingabe

Abbildung 1 : Systemumgebung für eine lokale PIN-Eingabe

Betrachtet wird in diesem Unterkapitel der Fall, dass eine Karte in einem eHKT steckt (in Abbildung 1 ist das ein HBA) und dass diese Karte durch PIN-Eingabe freigeschaltet wird. Der Unterschied zur "Remote PIN-Eingabe" in Kapitel 1.1.2 ist, dass:

1. die PIN-Eingabe am selben eHKT (lokal) vorgenommen wird, in welchem auch die freizuschaltende Karte steckt und
2. die eingegebene PIN das eHKT nie verlässt.

Technisch wird die Freischaltung einer Karte durch „lokale PIN-Eingabe" wie folgt bewerkstelligt: Die Komponente, welche das eHKT ansteuert (etwa ein Konnektor) sendet an das eHKT das Kommando `SICCT PERFORM VERIFICATION` gemäß [SICCT#5.19]. Diesem Kommando wird unter anderem als Parameter übergeben, auf welchen Kartenslot sich das Kommando bezieht. Im Wesentlichen bewirkt das Kommando `SICCT PERFORM VERIFICATION` dann folgendes:

1. Der Benutzer wird zur Eingabe einer PIN aufgefordert.
2. Die eingegebene PIN wird in das Kartenkommando "Command-To-Perform" eingebettet, welches als Parameter im Kommando `SICCT PERFORM VERIFICATION` enthalten ist. Im Falle der "lokalen PIN-Eingabe" handelt es sich beim "Command-To-Perform" um das Kommando `VERIFY` gemäß [gemSpec_COS#(N082.200)].
3. Das so ergänzte Kartenkommando wird an die Karte gesendet, dessen Slot in einem Eingangsparameter des Kommandos `SICCT PERFORM VERIFICATION` adressiert wurde (hier HBA).
4. Im Erfolgsfall meldet die Karte nur das Statuswort = ‘9000‘ = NoError, da für das Kommando `VERIFY` gemäß [gemSpec_COS#(N082.200)] keine Antwortdaten definiert sind.

5. Das eHKT beantwortet das Kommando `SICCT PERFORM VERIFICATION` dann ebenfalls nur mit dem Statuswort = '9000' = NoError.

Wichtig: Die Antwort auf das Kommando `SICCT ERFORM VERIFICATION` enthält nichts außer dem Statuswort.

1.1.2 Remote PIN-Eingabe

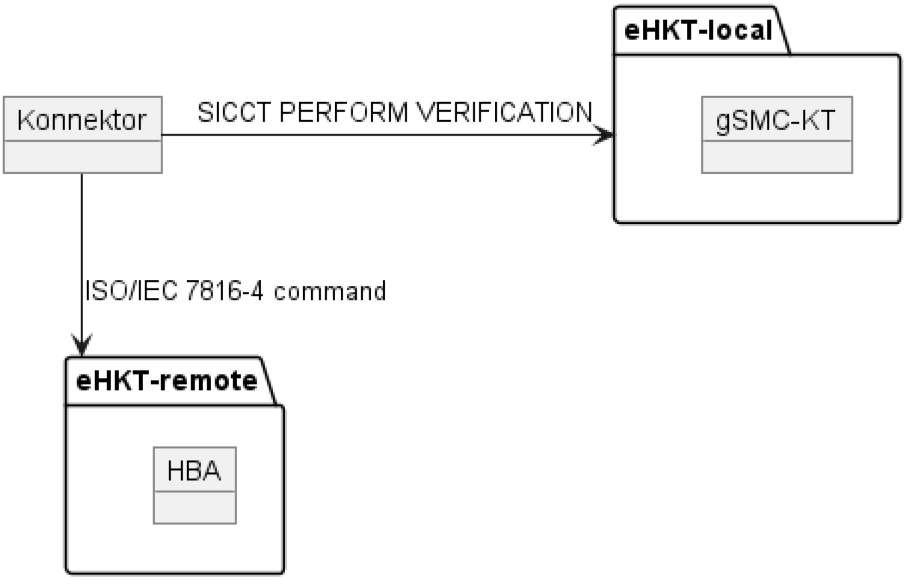

Abbildung 2 :Systemumgebung für eine "Remote PIN-Eingabe"

Betrachtet wird in diesem Unterkapitel der Fall, dass eine Karte in einem entfernten (remote) eHKT steckt (in Abbildung 2 ist das eine HBA) und dass diese Karte durch PIN-Eingabe freigeschaltet wird. Der Unterschied zur "lokalen PIN-Eingabe" aus Kapitel 1.1.1 ist, dass:

1. die PIN-Eingabe (möglicherweise) nicht am selben eHKT (lokal) vorgenommen wird, in welchem auch die freizuschaltende Karte (hier HBA) steckt oder
2. die eingegebene PIN die sichere Umgebung des eHKT verlässt.

Hinweis: Der Vollständigkeitshalber sei erwähnt, dass das hier beschriebene "Remote PIN-Verfahren" auch dann funktioniert, wenn „eHKT-remote" und „eHKT-local" identisch sind.

Technisch wird die Freischaltung einer Karte durch „remote" PIN-Eingabe wie folgt bewerkstelligt: Die Komponente, welche die eHKT ansteuert (etwa ein Konnektor) führt folgende Aktionen aus:

1. Aufbau eines kryptographisch gesicherten Kanals zwischen der gSMC-KT im „eHKT-local" und dem HBA im „eHKT-remote". Details spielen hier keine Rolle.
2. Kommando `SICCT PERFORM VERIFICATION` an das „eHKT-local" mit der gSMC-KT als Zielkarte. Der weitere Ablauf im „eHKT-local" ähnelt auf dieser Abstraktionsebene dem Ablauf aus Abbildung 1:
   a. Der Benutzer wird zur Eingabe einer PIN aufgefordert.

b. Die PIN wird in ein passendes Kartenkommando eingebettet. In diesem Fall handelt es sich beim "Command-To-Perform" um ein `PSO Encipher` Kommando gemäß [gemSpec_COS#(N091.446)].

c. Das so ergänzte Kartenkommando wird an die Karte gesendet, dessen Slot in einem Eingangsparameter adressiert wurde (hier gSMC-KT).

d. Anders als in Kapitel 1.1.1 wird das "Command-To-Perform" in diesem Fall Antwortdaten enthalten: Die verschlüsselte PIN.

e. Im Erfolgsfall liegt dann eben nicht nur ein Statuswort vor, sondern ein Statuswort plus eine verschlüsselte PIN.

f. Im Gutfall beantwortet das eHKT das Kommando `SICCT PERFORM VERIFICATION` dann mit der verschlüsselten PIN als Datenfeld der Antwort plus dem Statuswort.

3. Der Konnektor wird die verschlüsselte PIN in ein passendes, kryptographisch gesichertes `VERIFY` Kommando einbetten und an „eHKT-remote" mit dem HBA als Zielkarte gesendet.

4. Der HBA entschlüsselt das kryptographisch gesicherte Kommando, führt einen PIN-Vergleich durch und wird (im Gutfall) freigeschaltet.

Wichtig: Die Antwort auf das Kommando `SICCT PERFORM VERIFICATION` enthält zusätzlich zum Statuswort eine verschlüsselte PIN.

1.1.3 Kommando SICCT PERFORM VERIFICATION

Die generelle Struktur aller SICCT-Kommandos ist in [SICCT#5.1] beschrieben. Diese Struktur basiert auf [7816-4]. Demzufolge bestehen Kommandos aus folgenden Teilen:

1. Kommandoheader, zeigt unter anderem an, um welches Kommando es sich handelt,
2. optionales Datenfeld mit weiteren Kommandoparametern,
3. optionales Le-Feld, gibt die maximale Anzahl der Bytes im Datenfeld einer Antwort an. Fehlt das Le-Feld, dann enthält die zum Kommando korrespondierende Antwort niemals ein Datenfeld, sondern nur das Statuswort.

In diesem Zusammenhang ist lediglich interessant, dass [SICCT#5.19.3] für das Le-Feld zwei Varianten vorsieht:

1. Es ist zulässig, dass das Le-Feld fehlt.
2. Es ist zulässig, dass das Le-Feld vorhanden ist.

Für den in Kapitel 1.1.1 dargestellten Fall der "lokalen PIN-Eingabe" ist es unerheblich, ob das Kommando `SICCT PERFORM VERIFICATION` ein Le-Feld enthält oder nicht, weil der Use Case der lokalen PIN-Eingabe ohne das Datenfeld in der Antwort funktioniert.

Für den in Kapitel 1.1.2 dargestellten Fall der "remote PIN-Eingabe" ist es erforderlich, dass bei standardkonformer Implementierung des eHKT ein Le-Feld im Kommando `SICCT PERFORM VERIFICATION` vorhanden ist, weil der Use Case der "remote PIN-Eingabe" zwingend ein Datenfeld in der Antwort benötigt.

© gematik – öffentlich_Entwurf

1.2 Interoperabilitätsproblem

1.2.1 Definitionen

eHKT_good: So ein eHKT ist in Bezug auf das Kommando `SICCT PERFORM VERIFICATION` standardkonform in dem Sinne, dass es bei fehlendem Le-Feld nie Antwortdaten schickt (sondern nur das Statuswort).

eHKT_bad: So ein eHKT ist in Bezug auf das Kommando `SICCT PERFORM VERIFICATION` „großzügig" mit dem Le-Feld. So ein eHKT schickt auch bei fehlendem Le-Feld die passenden Antwortdaten im Falle des "remote PIN-Eingabe" Use Cases.

Konnektor_good: So ein Konnektor schickt im Kommando `SICCT PERFORM VERIFICATION`

1. nie ein Le-Feld, wenn "lokale PIN-Eingabe" vorliegt,
2. ein Le-Feld, wenn "remote PIN-Eingabe" vorliegt.

Konnektor_bad: So ein Konnektor schickt im Kommando `SICCT PERFORM VERIFICATION` nie ein Le-Feld.

1.2.2 Interoperabilitätstest

Im Rahmen von Interoperabilitätstests stellte sich heraus, dass:

1. Konnektor_good sowohl mit eHKT_good, als auch mit eHKT_bad wie gewünscht interagiert.
2. eHKT_bad sowohl mit Konnektor_good, als auch mit Konnektor_bad wie gewünscht interagiert.
3. Konnektor_bad lediglich mit eHKT_bad interagiert.

Diese Ergebnisse sind in Tabelle 1 zusammengefasst: Nur wenn sowohl Konnektor als auch eHKT im hier betrachteten Fall nicht standardkonform arbeiten ("bad" sind), dann sind Interoperabilitätsprobleme im Use Case "Remote PIN-Eingabe" zu erwarten.

Tabelle 1 : Kombinatorischer Überblick

		Konnektor	
		good	bad
eHKT	good	☺	X
	bad	☺	☺

1.3 Remote-PIN Sender und Empfänger

Gemäß Abbildung 2 und der zugehörigen Beschreibung in Kapitel 1.1.2 wird in den gematik Spezifikationen im Use Case "Remote PIN-Eingabe" zwischen Remote-PIN-

© gematik – öffentlich_Entwurf

Sender (RPS) und Remote PIN-Empfänger (RPE) unterschieden. Aus kryptographischer Sicht wird dabei die PIN im Remote-PIN-Sender verschlüsselt, dann zum Remote-PIN-Empfänger übertragen und dort entschlüsselt. Es besteht ein kryptographisch gesicherter Kanal (Trusted Channel) zwischen Remote-PIN-Sender und Remote-PIN-Empfänger.

Stand heute (Dezember 2022) gilt:

1. Passwortobjekte auf Karten der TI sind typischerweise so konfiguriert, dass PINs sowohl im Klartext (kryptographisch ungesichert), als auch im Trusted Channel (kryptographisch gesichert) zur Karte übertragbar sind. Eine Ausnahme bildet die PIN.QES des HBA für Stapel- und Komfortsignatur: Für Stapel- und Komfortsignatur ist die kryptographisch gesicherte PIN-Übergabe an den HBA erforderlich. Der HBA ist dann Remote-PIN-Empfänger. Nur die gSMC-KT ist dann ein geeigneter Remote-PIN-Sender.
2. Wird eine gSMC-KT als Remote-PIN-Sender eingesetzt, dann funktioniert die "remote PIN-Eingabe" nur gegenüber einem HBA oder einer SMC-B, weil nur HBA und SMC-B passende Identitäten besitzen, mit denen sich die Remote-PIN-Sender Funktionalität einer gSMC-KT nutzen lässt.
3. Die "Remote PIN-Eingabe" mit gSMC-KT einerseits und HBA oder SMC-B andererseits funktioniert mit beliebigen Passwortobjekten von HBA und SMC-B.
4. Der Aufbau des Trusted Channels zwischen Remote-PIN-Sender und Remote-PIN-Empfänger ist in [gemSpec_COS#CosA_7f8] beschrieben. Mit der Nomenklatur aus [gemSpec_COS#CosA_7f8 gilt:
   a. "COSa, elcSK4SM" ist der Remote-PIN-Empfänger (also HBA oder SMC-B).
   b. "COSb, elcSK4TC" ist der Remote-PIN-Sender (also die gSMC-KT).
   c. "Steuersoftware" ist im Wirkbetrieb ein Konnektor.

© gematik – öffentlich_Entwurf

2 Änderungen an gemSpec_KT

2.1 Konzept

Das Konzept für die Behandlung des Kommandos `SICCT PERFORM VERIFICATION` sieht folgendes vor:

1. Zusätzlich zu den bereits bestehenden Festlegungen bezüglich `SICCT PERFORM VERIFICATION` in [gemSpec_KT] und [SICCT] wird in [gemSpec_KT] das Verhalten bezüglich des Le-Feldes dieses SICCT-Kommandos genau festgelegt.
2. Diesbezügliche Anforderungen richten sich an diejenige Komponente, welche ein eHKT nutzt (typischerweise ein Konnektor oder auch ein Testsystem). Diese Komponenten dürfen ein eHKT dann nur im Rahmen dieser Anforderungen ansteuern. Verletzen diese Komponenten diese Anforderungen, dann wird das eHKT außerhalb seiner Spezifikation betrieben und das Verhalten ist aus funktionaler Sicht irrelevant (aus Sicherheitssicht darf ein Betreiben außerhalb der Spezifikation natürlich nicht bedenklich sein).
3. Fehlt im Kommando `SICCT PERFORM VERIFICATION` das Le-Feld, dann wird dem eHKT der Use Case "lokale PIN-Eingabe" signalisiert.
4. Enthält das Kommando `SICCT PERFORM VERIFICATION` ein Le-Feld, dann wird dem eHKT der Use Case "remote PIN-Eingabe" signalisiert.

Hinweise zu erwartetem Verhalten:

1. *Eine gSMC-KT, die sich strikt an [gemSpec_gSMC-KT_ObjSys_G2.1] orientiert besitzt keine Passwortobjekte. Deshalb werden dann alle Use Cases zur "lokalen PIN-Eingabe" von der gSMC-KT mit dem Fehlercode "6A88" = PasswordNotFound beantwortet.*
2. *Es ist denkbar eine Test gSMC-KT mit einem Passwortobjekt auszustatten. Dann ist die Testerwartungshaltung, dass der Use Case "lokale PIN-Eingabe" auch für so eine gSMC-KT funktioniert.*
3. *Der Use Case "remote PIN-Eingabe" ist nur dann erfolgreich durchführbar, wenn die Zielkarte des lokalen eHKT (also die Zielkarte für das Kommando `SICCT PERFORM VERIFICATION` mit vorhandenem Le-Feld) einen Trusted Channel mit Sessionkey4TC etabliert hat. Ist das nicht der Fall, dann wird die Zielkarte mit einer Fehlermeldung antworten, wenn das eHKT versucht mit so einer Zielkarte eine kryptographische Sicherung der Kommando APDU zu bewerkstelligen.*
4. *Testweise ist es denkbar eine gSMC-K (= Gerätekarte Konnektor) zusätzlich in ein eHKT zu stecken oder eGK, HBA oder SMC-B mit passendem Schlüsselmaterial so auszustatten, dass diese Karten einen Trusted Channel mit Sessionkey4TC aufbauen. Es wird erwartet, dass die Implementierung des Kommandos `SICCT PERFORM VERIFICATION` im eHKT so generisch ist, dass derartige Karten dann auch als Zielkarte eines Kommandos `SICCT PERFORM VERIFICATION` mit Le-Feld vorhanden fungieren.*
5. *Bereits [SICCT] legt fest, dass das Kommando `SICCT PERFORM VERIFICATION` mit dem Fehlercode der Zielkarte zu antworten hat, wenn die Zielkarte einen Fehler meldet. Deshalb ist es nicht erforderlich im Rahmen dieses Dokumentes auf Fehlerfälle einzugehen.*

© gematik – öffentlich_Entwurf

196

2.2 Neue Anforderungen

In [gemSpec_KT] werden am Ende des Kapitels 3.7.5 "Ergänzungen des Kommandos SICCT PERFORM VERIFICATION" folgende Anforderungen und folgender Hinweis neu aufgenommen:

A_22869 - K_externeWelt, SICCT PERFORM VERIFICATION, lokale PIN-Eingabe
Der Nutzer eines Terminals DARF in das Kommando `SICCT PERFORM VERIFICATION` KEIN Le-Feld einstellen, falls der Use Case "lokale PIN-Eingabe" vorliegt. In diesem Fall MUSS das "Command-To-Perform" ein `VERIFY` Kommando gemäß [gemSpec_COS#(N082.200)] sein.**[<=]**

A_22871 - K_externeWelt: SICCT PERFORM VERIFICATION, remote PIN-Eingabe
Der Nutzer eines Terminals MUSS in das Kommando `SICCT PERFORM VERIFICATION` ein Le-Feld einstellen, falls der Use Case "remote PIN-Eingabe" vorliegt. In diesem Fall MUSS das "Command-To-Perform" ein `PSO Encipher` Kommando gemäß [gemSpec_COS#(N091.446)] sein. Das Le-Feld (siehe [gemSpec_COS#11.6.1]) MUSS den Wert:
a. WildCardShort besitzen, falls Nc <= 255 oder,
b. WildCardExtended besitzen, falls Nc > 255 ist.**[<=]**

Hinweis: Nc gibt die Anzahl Oktette im (optionalen) Datenfeld einer Kommando-APDU (hier `SICCT PERFORM VERIFICATION`) an, vergleiche [gemSpec_COS#11.5.5].

Hinweis: Für die kryptographische Sicherung einer Kommando APDU sind folgende Use Cases relevant:

1. *PSO Encipher gemäß [gemSpec_COS#(N091.446)] für die Verschlüsselung der PIN und*
2. *PSO Verify Cryptographic Checksum gemäß [gemSpec_COS#(Nß87.228)] für die MAC-Berechnung.*

© gematik – öffentlich_Entwurf

3 Referenzierte Dokumente

Quelle	Herausgeber
[7816-4]	ISO/IEC 7816 Identification card - Integrated circuit cards, Part 4: Organization, security and commands for interchange, May2020
[SICCT]	Secure Interoperable Chip Card Terminal, Version 1.2.3, Stand 30.09.2016, siehe https://www.teletrust.de/fileadmin/docs/projekte/sicct/SICCT-Spezifikation_V_1.2.3_vom_30.09.2016.pdf

© gematik – öffentlich_Entwurf