1 Einleitung

Gemäß den Festlegungen in [gemSpec_KT] lässt sich ein eHealth-Kartenterminal nur dann sinnvoll nutzen, wenn es mit einem TLS-Client verbunden ist, der sich im Rahmen des TLS-Aufbaus (vergleiche RFC 5246) erfolgreich als "Konnektor" authentisiert hat. In diesem Zusammenhang definiert [gemSpec_KT#A_22456] die Zustände*State_ClientWithoutPairing* und *State_ClientWithPairing*.

Der dort ebenfalls beschriebene Zustand *State_InvalidClient* ist für Anwendungsfälle gemäß [gemSpec_KT#CMD_KT_0004] wichtig, unter anderem für den Download und die Installation einer neuen eHealth-Kartenterminal Firmware.

Daraus folgt, dass einerseits der erfolgreiche Aufbau einer TLS-Verbindung zu beliebigen Clients wichtig ist (auch, ohne dass diese Clients sich authentisieren), und andererseits auch zu Clients, die ihre Rolle als Konnektor nachgewiesen haben. Es ist denkbar, dass der Nachweis "Rolle als Konnektor" aus vielerlei Gründen scheitert, beispielsweise:

1. Client präsentiert kein Zertifikat,
2. Client präsentiert kein End-Entity Zertifikat,
3. Client präsentiert ein End-Entity Zertifikat welches:
    a. sich wegen eines fehlenden CA-Schlüssels nicht prüfen lässt,
    b. nicht konform zum X.509 Standard ist,
    c. nicht die Rolle "Konnektor" enthält,
    d. eine ungültige Signatur enthält
    e. . . .
4. Client präsentiert eine valide Zertifikatskette mit einem CA-Zertifikat, welches nicht in der dafür vorgesehenen TSL enthalten ist.
5. . . .

Typischerweise werden zum Aufbau einer TLS-Verbindung vielfach bewährte Standardbibliotheken (etwa OpenSSL) verwendet. Derartige Standardbibliotheken führen eine Reihe von Prüfungen beim TLS-Verbindungsaufbau durch. Einige Prüfungen lassen sich durch entsprechende Konfigurationsparameter ausschalten. Insbesondere die mathematische Prüfung des X.509 Zertifikates (siehe [gemSpec_KT#TIP1-A_3933) lässt sich bei OpenSSL allerdings nicht abschalten.

Derzeit legt [gemSpec_KT#TIP1-A_3095-01] inklusive der erläuternden Hinweise (hier Unterpunkt 3) fest, dass auch bei fehlerhafter Signatur des X.509 Zertifikates ein TLS-Verbindungsaufbau zustande kommen muss.

Daraus folgt, dass bei spezifikationskonformer Umsetzung von [gemSpec_KT#TIP1-A_3095-01] die Bibliothek OpenSSL so nicht verwendbar wäre.

Es wäre denkbar, den Code von OpenSSL so anzupassen, dass die derzeitige Version von [gemSpec_KT#TIP1-A_3095-01] voll erfüllt wird. Da derartige Änderungen sowohl aus funktionaler Sicht als auch aus Sicherheitssicht kritisch und nicht trivial sind, wird eine derartige Anpassung nicht favorisiert. Bei jeder Weiterentwicklung oder Bugfixing in OpenSSL wäre eine solche Änderung erneut durchzuführen. Das erscheint fehleranfällig.

Attraktiver erscheint folgendes Konzept: Die Anforderungslage in [gemSpec_KT] wird dahingehend geändert, dass genauer festgelegt wird, wann eine TLS-Verbindung in

jedem Fall zustande kommen muss. Alle anderen Fälle werden bewusst nicht spezifiziert. Damit liegt es im Belieben des KT-Herstellers das Verhalten in den übrigen Fällen festzulegen.

Dabei wird folgender Ansatz verfolgt:

1. Führt ein Client den TLS-Handshake gemäß [gemSpec_Krypt] aus ohne ein Clientzertifikat zu präsentieren, dann kommt eine TLS-Verbindung zu Stande und das eHealth-KT befindet sich anschließend im Zustand *State_InvalidClient*.
2. Führt ein Client den TLS-Handshake gemäß [gemSpec_Krypt] aus und präsentiert dabei das "perfektes" X.509 Zertifikat mit der Rolle "Konnektor" und authentisiert sich erfolgreich, dann kommt die TLS-Verbindung zu Stande und das eHealth-KT befindet sich anschließend je nach Pairing-Situation in einem der Zustände *State_ClientWithoutPairing* oder *State_ClientWithPairing*.
3. Alle anderen Fälle werden in [gemSpec_KT] nicht betrachtet.

Was jetzt noch fehlt, ist die Festlegung wann ein X.509 Zertifikat "perfekt" ist. Dazu verweist [gemSpec_KT] zukünftig auf die Zertifikatsprüfung gemäß [gemSpec_PKI#TUC_PKI_018], schränkt allerdings ein, dass einzelne Schritte an die Gegebenheiten des Kartenterminals anzupassen seien und listet die erforderlichen Prüfpunkte vollständig auf.

© gematik – öffentlich_Entwurf

2 Änderungen an gemSpec_KT

Die Kapitel 3.10 und 3.11 inklusive aller Unterkapitel werden ersetzt. Änderungen sind im Folgenden gelb markiert. Dabei gilt folgendes:

1. Neuer Text ist gelb hinterlegt.
2. ~~Entfallener Text ist durchgestrichen und gelb hinterlegt.~~
3. Unveränderter Text wird nicht gelb hinterlegt, auch dann nicht, wenn er innerhalb des Dokumentes lediglich verschoben wird.

3.10 Aufbau der SICCT-spezifischen TLS-Verbindungen

Der Ablauf des TLS-Verbindungsaufbaus zwischen einem TLS-Client und dem eHealth-Kartenterminal ist im folgenden Diagramm „Pic_KT_0016 TLS-Verbindungsaufbau" informativ dargestellt.

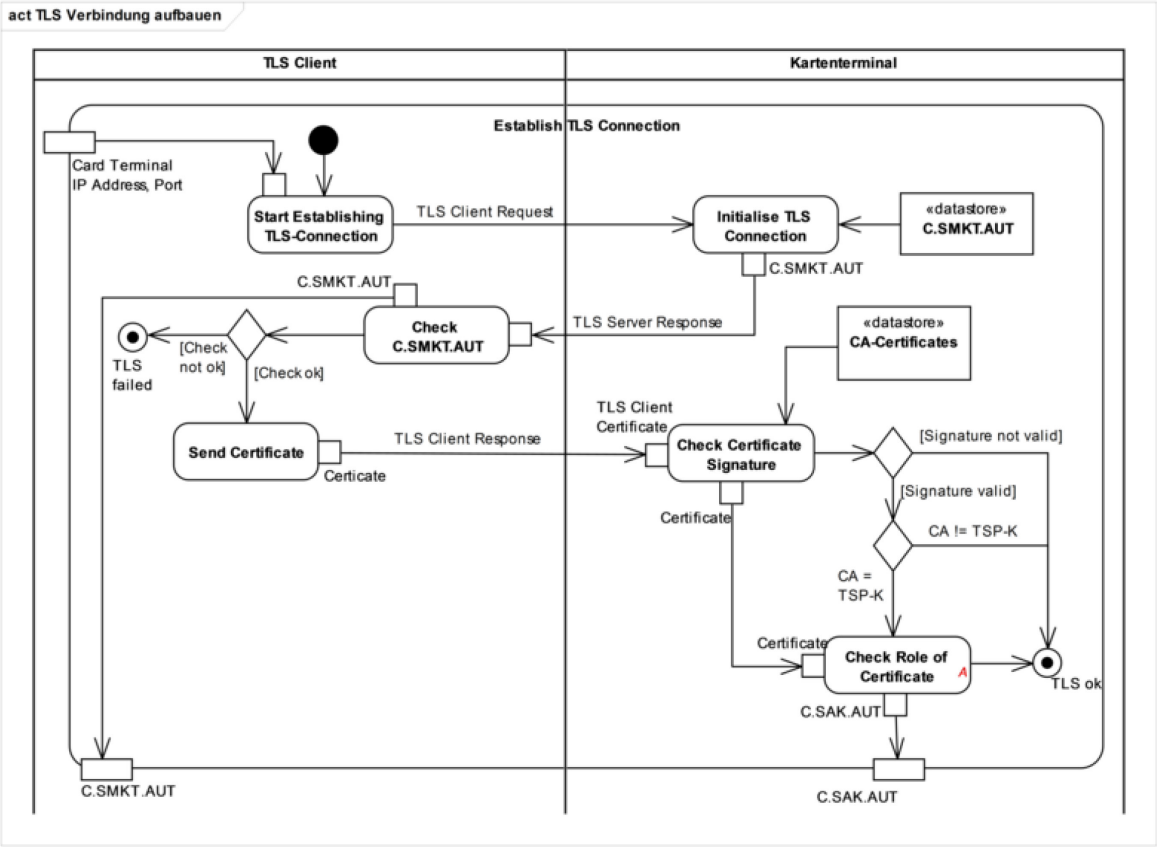

Abbildung 1: Pic_KT_0016 TLS-Verbindungsaufbau

Für den Aufbau einer TLS-Verbindung im Rahmen eines TLS-Handshakes existieren in den einschlägigen Industrie- und Internetstandards (etwa [RFC-5246#7.3]) eine Reihe von Optionen. Für einen interoperablen Betrieb zwischen Client (etwa Konnektor) und Server (hier eHealth-Kartenterminal) werden Festlegungen zum TLS-Handshake in [gemSpec_Krypt] getroffen.

Dort und weiter unten in diesem Dokument wird unter anderem festgelegt, dass ein eHealth-Kartenterminal nur dann fachlich nutzbar ist, wenn sich der TLS Client authentisiert. Dazu ist es erforderlich, dass der TLS Client im Rahmen einer Client Ceritifcate Nachricht (siehe [RFC5246#7.4.6]) ein End-Entity Zertifikat präsentiert. Es ist möglich, dass eine Client Certificate Nachricht eine komplette Zertifikatskette enthält. Es wird gefordert, dass ein darin enthaltenes End-Entity Zertifikat mit einem CA-Zertifikat geprüft wird, welches bereits vor dem TLS-Handshake im eHealth-Kartenterminal gespeichert war.

Die folgenden Kapitel legen Einzelheiten zur Zertifikatsprüfung fest.

3.10.1 Speicherung von CA-Zertifikaten

TIP1-A_3255 - CA-Zertifikate der relevanten TSP speichern
Das eHealth-Kartenterminal MUSS mindestens die CA-Zertifikate der TSP aus der integeren und authentischen TSL speichern (z. B. in der Firmware), die Komponentenzertifikate für einen Konnektor erzeugen.
[<=]

Hinweis: Die Dienste bzw. CA-Zertifikate in der TSL lassen über die TSL-Extension zuordnen: Im Extensioneintrag wird zu jedem CA-Zertifikat angegeben, welche Typen von Zertifikaten es ausstellen darf (siehe [gemSpec_TSL#7.3.2.1]). Ein Filtern nach relevanten TSPs ist damit einfach möglich, gemäß A_22889 aber nicht erforderlich.

TIP1-A_3256 - CA-Zertifikate in Kartenterminal und anschließende Speicherung
Das eHealth-Kartenterminal MUSS beim Einbringen von CA-Zertifikaten in das Kartenterminal und ihrer anschließenden Speicherung innerhalb des Kartenterminals deren Authentizität gewährleisten.
[<=]

TIP1-A_3257 - Schutz CA-Zertifikate
Das eHealth-Kartenterminal MUSS gespeicherte CA-Zertifikate gegen Veränderungen schützen.
[<=]

TIP1-A_6482 - Anzahl CA-Zertifikate
Das eHealth-Kartenterminal MUSS zu einem Zeitpunkt mindestens zehn CA-Zertifikate pro Vertrauensraum speichern können.
[<=]

Hinweis: Wenn zeitgleich mehrere verschiedene Vertrauensräume in der Firmware des eHealth-Kartenterminals hinterlegt sind, so ist die Anzahl entsprechend zu vervielfachen.

TIP1-A_3094 - Aktualisierung von CA-Zertifikaten der Komponenten-PKI
Nehmen neue CAs ihren Betrieb für das Generieren von Komponentenzertifikaten für Konnektoren auf, MUSS das eHealth-Kartenterminal die zugehörigen CA-Zertifikate auf vertrauenswürdige Weise übernehmen.
[<=]

TIP1-A_3941 - Update von TSP-Zertifikaten
Der Hersteller eines eHealth-Kartenterminals KANN zur Umsetzung von [TIP1-A_3094] das Update von TSP-Zertifikaten über ein Update der Firmware des Kartenterminals realisieren.
[<=]

TIP1-A_3158 - TSP-Update-Mechanismus
Hersteller KÖNNEN zur Umsetzung von [TIP1-A_3094] einen TSP-Update-Mechanismus am eHealth-Kartenterminal implementieren, welcher es ermöglicht, die Liste der TSP CAs auszutauschen.
[<=]

TIP1-A_3159 - TSP-Update-Mechanismus für KT ohne Firmware-Update
Ein eHealth-Kartenterminal, das den TSP-Update-Mechanismus gemäß [TIP1-A_3158] umsetzt, DARF für diesen ein Firmware-Update NICHT nutzen bzw. erforderlich machen.
[<=]

Hinweis: Die Sicherheit des TSP-Update-Mechanismus ist im Rahmen der Common Criteria Evaluierung nachzuweisen. Die Details zur Umsetzung sind herstellerspezifisch.

© gematik – öffentlich_Entwurf

3.10.2 Prüfen eines Client-Zertifikates

Komponentenzertifikate für Konnektoren werden durch Trusted Service Provider für Komponentenzertifikate (TSP) ausgestellt. Jedes Komponentenzertifikat eines Konnektors lässt sich auf ein CA-Zertifikat innerhalb der Trust-Service Status List (TSL) zurückführen. Es ist dabei durch organisatorische Prozesse im Rahmen der Baureihenzulassung sichergestellt, dass nur betriebszugelassene Geräte mit solchen Komponentenzertifikaten für Konnektoren ausgestattet werden.

Zur Prüfung eines Client-Zertifikates wird der Use Case [gemSpec_PKI#TUC_PKI_018] verwendet, wobei einige Schritte an die Gegebenheiten des eHealth-Kartenterminals angepasst werden.

[gemSpec_PKI#TUC_PKI_018] definiert folgende Eingangsdaten für die "Zertifikatsprüfung in der TI":

1. Zertifikat
2. Referenzzeitpunkt
3. PolicyList
4. intendedKeyUsage
5. intendedExtendedKeyUsage
6. OCSP-Graceperiod (Default: 10 Minuten)
7. Offline-Modus aus der Menge {ja, nein}
8. Beigefügte OCSP-Response (optional)
9. Timeout-Parameter (Default: 10 Sekunden)
10. TOLERATE_OCSP_FAILURE aus der Menge {true, false}
11. Prüfmodus aus der Menge {OCSP, CRL}

[gemSpec_PKI#TUC_PKI_018] beschreibt folgenden Standardablauf für die "Zertifikatsprüfung in der TI":

1. [gemSpec_PKI#TUC_PKI_002], Gültigkeit prüfen
2. Prüfen der keyUsage aus dem Zertifikat gegen die Eingangsdaten "intendedKeyUsage" und "intendedExtendedKeyUsage"
3. [gemSpec_PKI#TUC_PKI_003], passendes CA-Zertifikat suchen
4. [gemSpec_PKI#TUC_PKI_004], mathematische Prüfung der Zertifikatssignatur
5. ServiceStatus des CA-Zertifikates prüfen
6. Falls "Offline-Modus == ja" ist, dann Schritt 7 überspringen
7. (nur relevant für "Prüfmodus == OCSP" und "Offline-Modus == nein"), [gemSpec_PKI#TUC_PKI_006], OCSP-Prüfung des Zertifikates
8. [gemSpec_PKI#TUC_PKI_009], Rollenermittlung
9. [gemSpec_PKI#TUC_PKI_007], Prüfung Zertifikatstyp

Hinweis zu Prüfschritt 1: Möglicherweise verfügt ein eHealth-Kartenterminal nicht über zuverlässige Informationen zu Datum und Uhrzeit. Dann ist es gemäß A_22881 zulässig, Zertifikate auch dann zu akzeptieren, wenn deren Gültigkeit abgelaufen ist.

Hinweis zu Prüfschritt 3: Gemäß A_22888 werden CA-Zertifikate ausschließlich im Speicher des eHealth-Kartenterminals gesucht und niemals einer Client Certificate Nachricht entnommen.

Hinweis zu Prüfschritt 6: Möglicherweise verfügt ein eHealth-Kartenterminal über keine Verbindung zu einem OCSP-Responder. Dann ist es gemäß A_22883 zulässig im Prüfmodus "CRL" gegen eine leere Certificate Revocation List zu prüfen. In diesem Fall akzeptiert das eHealth-Kartenterminal auch gesperrte oder revozierte End-Entity Zertifikate.

Definition: Ein Zertifikat wird im Folgenden als "perfekt" bezeichnet, wenn es alle Prüfungen gemäß [gemSpec_PKI#TUC_PKI_018] mit folgenden Eingangsdaten besteht:

1. Zertifikat = End-Entity Zertifikat aus der Client Certificate Nachricht
2. Referenzzeitpunkt = Systemzeit = aktuelles Datum und aktuelle Uhrzeit
3. PolicyList = {sak-aut} = {1.2.276.0.76.4.113}
4. intendedKeyusage = {digitalSignature}, (siehe RFC5280#4.2.1.3)
5. intendedExtendedKeyUsage = {tlsClientAuthentication} = {1.3.6.1.5.5.7.3.2}
6. OCSP-Graceperiod = 10 Minuten oder kleiner
7. Offline-Modus = nein
8. Beigefügte OCSP-Response = (leer)
9. Timeout-Parameter = 10 Sekunden oder kleiner
10. TOLERATE_OCSP_FAILURE = false
11. Prüfmodus = OCSP

Die gematik erlaubt es einem eHealth-Kartenterminal im Rahmen des Use Cases [gemSpec_PKI#TUC-PKI_018] jedes Zertifikat als gültig anzusehen, falls das eHealth-Kartenterminal über keine zuverlässige Information bezüglich Datum und Uhrzeit verfügt.

Definition: Ein Zertifikat wird im Folgenden als "fuzzyValidity" bezeichnet, wenn es alle Prüfungen gemäß [gemSpec_PKI#TUC_PKI_018] besteht, ohne dass ein zuverlässiger Referenzzeitpunkt vorliegt (weshalb hier eine Unterscheidung zwischen "perfekt" und "ungültig" nicht möglich ist). Das entspricht folgenden Eingangsdaten:

1. Zertifikat = End-Entity Zertifikat aus der Client Certificate Nachricht
2. Referenzzeitpunkt = beliebig aus dem Intervall [notBefore, notAfter]
3. PolicyList = {sak-aut} = {1.2.276.0.76.4.113}
4. intendedKeyusage = {digitalSignature}, (siehe RFC5280#4.2.1.3)
5. intendedExtendedKeyUsage = {tlsClientAuthentication} = {1.3.6.1.5.5.7.3.2}
6. OCSP-Graceperiod = 10 Minuten oder kleiner
7. Offline-Modus = nein
8. Beigefügte OCSP-Response = (leer)
9. Timeout-Parameter = 10 Sekunden oder kleiner
10. TOLERATE_OCSP_FAILURE = false
11. Prüfmodus = OCSP

© gematik – öffentlich_Entwurf

Die gematik erlaubt es einem eHealth-Kartenterminal im Rahmen des Use Cases [gemSpec_PKI#TUC-PKI_018] den Prüfmodus "CRL" zu wählen und dabei offline gegen eine leere Certificate Revocation List zu prüfen, falls es weder über eine Verbindung zu einem OCSP-Responder noch über eine Certification Revocation List verfügt.

Definition: Ein Zertifikat wird im Folgenden als "fuzzyRevocation" bezeichnet, wenn es alle Prüfungen gemäß [gemSpec_PKI#TUC_PKI_018] im Offline-Modus gegen eine leere Certificate Revocation List besteht (weshalb hier eine Unterscheidung zwischen "perfekt" und "gesperrt oder revoziert" nicht möglich ist). Dies entspricht folgenden Eingangsdaten:

1. Zertifikat = End-Entity Zertifikat aus der Client Certificate Nachricht
2. Referenzzeitpunkt = Systemzeit = aktuelles Datum und aktuelle Uhrzeit
3. PolicyList = {sak-aut} = {1.2.276.0.76.4.113}
4. intendedKeyusage = {digitalSignature}, (siehe RFC5280#4.2.1.3)
5. intendedExtendedKeyUsage = {tlsClientAuthentication} = {1.3.6.1.5.5.7.3.2}
6. OCSP-Graceperiod = beliebig, da irrelevant
7. Offline-Modus = ja
8. Beigefügte OCSP-Response = (leer)
9. Timeout-Parameter = beliebig, da irrlevant
10. TOLERATE_OCSP_FAILURE = true
11. Prüfmodus = CRL mit leerer Certificate Revocation List

Die gematik erlaubt es einem eHealth-Kartenterminal im Rahmen des Use Cases [gemSpec_PKI#TUC-PKI_018]:

1. den Prüfmodus "CRL" zu wählen, dabei auch gegen eine leere Certificate Revocation List zu prüfen, falls es über keine Verbindung zu einem OCSP-Responder verfügt, und
2. jedes Zertifikat als gültig anzusehen, falls das eHealth-Kartenterminal über keine zuverlässige Information bezüglich Datum und Uhrzeit verfügt.

Definition: Ein eHealth-Kartenterminal MUSS ein Zertifikat als "fuzzyValidity + fuzzyRevocation" einstufen, wenn es alle Prüfungen gemäß [gemSpec_PKI#TUC_PKI_018] im Offline-Modus gegen eine leere Certificate Revocation List besteht, ohne das ein zuverlässiger Referenzzeitpunkt vorliegt (weshalb hier eine Unterscheidung zwischen "perfekt" einerseits und andererseits "ungültig" oder "gesperrt oder revoziert" nicht möglich ist). Dies entspricht folgenden Eingangsdaten:

1. Zertifikat = End-Entity Zertifikat aus der Client Certificate Nachricht
2. Referenzzeitpunkt = beliebig aus dem Intervall [notBefore, notAfter]
3. PolicyList = {sak-aut} = {1.2.276.0.76.4.113}
4. intendedKeyusage = {digitalSignature}, (siehe RFC5280#4.2.1.3)
5. intendedExtendedKeyUsage = {tlsClientAuthentication} = {1.3.6.1.5.5.7.3.2}
6. OCSP-Graceperiod = beliebig, da irrelevant
7. Offline-Modus = ja

© gematik – öffentlich_Entwurf

8. Beigefügte OCSP-Response = (leer)
9. Timeout-Parameter = beliebig, da irrlevant
10. TOLERATE_OCSP_FAILURE = true
11. Prüfmodus = CRL mit leerer Certificate Revocation List

Definition: Ein Zertifikat wird im Folgenden als "aus Sicht des eHealth-Kartenterminals perfekt" bezeichnet, wenn es "perfekt" oder im Sinne von A_22882 "fuzzyValidity" oder im Sinne von A_22883 "fuzzyRevocation" ist (einschließendes ODER).

A_22888 - K_KT, CA-Zertifikat suchen
Das eHealth-Kartenterminal DARF bei der Suche nach einem passenden CA-Zertifikat gemäß [gemSpec_PKI#TUC_PKI_003] NUR solche Zertifikate berücksichtigen, die:

1. im eHealth-Kartenterminal gespeichert sind und
2. berechtigt sind Konnektorzertifikate auszustellen.

[<=]

A_22889 - K_Hersteller-KT, Liste zulässiger CA-Zertifikate
Der Hersteller des eHealth-Kartenterminals MUSS organisatorisch sicherstellen, dass nur solche CA-Zertifikate bei der Suche gemäß A_22888 berücksichtigt werden, die berechtigt sind Konnektorzertifikate auszustellen, falls die Firmware eines eHealth-Kartenteminals die Rolle eines CA-Zertifikates nicht prüft. **[<=]**

3.10.3 TLS-Handshake

A_22890 - K_KT, erfolgreicher TLS-Handshake
Das eHealth-Kartenterminal MUSS den TLS-Handshake erfolgreich durchlaufen, wenn dieser gemäß [gemSpec_Krypt] ausgeführt wird und die Zertifikatsliste in der Client Certificate Nachricht:

1. leer ist, oder
2. ein "aus Sicht des eHealth-Kartenterminals perfektes" End-Entity Zertifikat enthält und der Client sich erfolgreich authentisiert.

[<=]

Hinweis: Bei leerer Zertifikatsliste ist das eHealth-Kartenterminal nach dem TLS-Handshake im Zustand State_InvalidClient, siehe A_22456.

Hinweis: Enthält die Zertifikatsliste ein "aus Sicht des eHealth-Kartenterminals perfektes" End-Entity Zertifikat (vergleiche A_22887), dann liegt nach dem TLS-Handshake (je nach Pairing-Information) einer der Zustände State_ClientWithoutPairing oder State_ClientWithPairing vor, siehe A_22456.

Hinweis: Das eHealth-Kartenterminal präsentiert im Rahmen des TLS-Handshake sein SMKT-Zertifikat (C.SMKT.AUT). Wenn die Zertifikatsprüfung für das SMKT-Zertifikat im Konnektor scheitert, dann bricht der Konnektor die Verbindung ab.

Hinweis: Falls die Zertifikatsprüfung für das SMKT-Zertifikat im Konnektor erfolgreich verläuft, dann wird dieser prüfen, ob für das SMKT-Zertifikat Pairing-Informationen vorliegen. Liegen keine Pairing-Informationen im Konnektor vor, dann führt der Konnektor nur die in CMD_KT_005 gelisteten Kommandos aus. Mit diesem eingeschränkten Funktionsumfang lassen sich Pairing-Informationen etablieren. Liegen Pairing-Informationen im Konnektor vor, dann prüft er diese gemäß 2.5.2.2.

~~**TIP1-A_3254-01 – Prüfung betriebszugelassener Konnektor**~~
~~Falls die Liste der ClientCertificate-Nachricht gemäß A_22455 ein End-Entity-Zertifikat enthält, dann MUSS das eHealth-Kartenterminal dieses daraufhin überprüfen, ob es von einem betriebszugelassenen Konnektor stammt. <=~~

~~**TIP1-A_3940 – Zertifikat prüfen**~~
~~Das eHealth-Kartenterminal MUSS, zur Feststellung gemäß [TIP1-A_3254], ob das ansteuernde System ein betriebszugelassener Konnektor ist, das vom Konnektor präsentierte Zertifikat prüfen. <=~~

~~Dabei können Teile des Use Cases TUC_PKI_018 [gemSpec_PKI#8.3.1.1] verwendet werden, wobei die einzelnen Schritte jedoch an die Gegebenheiten des Kartenterminals angepasst werden müssen. Für die Verifikation müssen die folgenden Punkte umgesetzt werden.~~

~~Für eine automatische Prüfung der Betriebszulassung eines Konnektors durch andere IT-Systeme steht ein X.509-Zertifikat zusammen mit den damit verbundenen geheimen und öffentlichen Schlüsseln im Rahmen der Identitäten des Konnektors zur Verfügung.~~

~~**TIP1-A_3933 – Mathematische Prüfung Zertifikat**~~
~~Das eHealth-Kartenterminal MUSS das beim TLS-Aufbau präsentierte Konnektorzertifikat entsprechend TUC_PKI_004 gemäß [gemSpec_PKI#8.3.1.4] prüfen. <=~~

~~**TIP1-A_3934 – Ermittlung Zertifikatsrolle**~~
~~Das eHealth-Kartenterminal MUSS aus dem beim TLS-Aufbau präsentierten Konnektorzertifikat entsprechend TUC_PKI_009 gemäß [gemSpec_PKI#8.3.3.2] die Rolle ermitteln. <=~~

~~**TIP1-A_3935 – Vergleich Zertifikatsrolle**~~
~~Das eHealth-Kartenterminal MUSS überprüfen, dass die in [TIP1-A_3934] ermittelte Rolle der Rolle "Signaturanwendungskomponente (SAK)" (oid_sak gemäß gemSpec_OID#3.5.4) entspricht. <=~~

~~**TIP1-A_4115 – Sicherstellung CA Berechtigung**~~
~~Im Rahmen der Prüfung nach [TIP1-A_3933] MUSS das eHealth-Kartenterminal sicherstellen, dass nur Zertifikate von CAs zur Prüfung herangezogen werden, die berechtigt sind, Konnektorzertifikate auszustellen. <=~~

~~**A_21812 – Datums- und Zeitinformation**~~
~~Das eHealth-Kartenterminal KANN über eine Systemuhr oder anderweitige Datums- und Zeitinformation verfügen. Falls das eHealth-Kartenterminal über eine Datums- und Zeitinformation verfügt, dann wird empfohlen, dass Zertifikate als ungültig eingestuft werden, wenn ihre zeitliche Gültigkeit abgelaufen ist. <=~~

~~Die folgende Tabelle zeigt die einzelnen Schritte, die durchgeführt werden müssen:~~

~~**Tabelle 1: Schritte beim Verifizieren des Zertifikats einer Signaturanwendungskomponente (SAK)**~~

~~Aufgabe~~	~~TUC gemäß [gemSpec_PKI]~~	~~Besonderheit~~
~~Gültigkeit des Zertifikats prüfen~~	~~–~~	~~Siehe Anmerkungen unten.~~

~~CA-Zertifikat der ausstellenden CA suchen~~	~~-~~	~~Muss anhand der gespeicherten CA-Zertifikate durchgeführt werden. Siehe Anmerkungen unten.~~
~~Prüfung der Signatur über das Zertifikat~~	~~TUC_PKI_004~~	~~-~~
~~Prüfung, ob CA-Zertifikate für Konnektoren ausstellen darf~~	~~-~~	~~Wird organisatorisch im Vorfeld oder technisch geregelt. Siehe Anmerkungen unten.~~
~~Ermittlung der Rolle des Zertifikats~~	~~TUC_PKI_009~~	~~Ausgabe: OID der Rolle~~
~~Abgleich der Rolle mit der technischen Rolle "Signaturanwendungskomponente (SAK)"~~	~~-~~	~~OID = oid_sak?~~

~~Anmerkungen:~~

1. ~~Die gematik betrachtet es als zulässig, wenn ein eHealth-Kartenterminal keine Systemuhr besitzt und auch anderweitig keine Information zu Datum oder Zeit zur Verfügung hat. Als Konsequenz folgt daraus, dass eHealth-Kartenterminals ohne Datums- und Zeitinformation nicht in der Lage sind die zeitliche Gültigkeit von Zertifikaten zu prüfen.~~

2. ~~Es muss die Liste der in dem eHealth-Kartenterminal intern gespeicherten CA-Zertifikate durchsucht werden (siehe [TIP1-A_3936]). Zu einem Komponentenzertifikat eines Konnektors erfüllt (nur) das korrekte CA-Zertifikat folgende Bedingungen (siehe auch TUC_PKI_003 in [gemSpec_PKI#8.3.1.3]):~~

~~issuerDN Komponentenzertifikat = subjectDN CA-Zertifikat~~

~~authorityKeyIdentifier Komponentenzertifikat = subjetKeyIdentifier CA-Zertifikat~~

- ~~Wird [TIP1-A_4115] nicht technisch im Kartenterminal umgesetzt, dann muss durch organisatorische Maßnahmen sichergestellt werden, dass nur für solche CAs die CA-Zertifikate in das eHealth-Kartenterminal eingebracht werden, die auch tatsächlich Komponentenzertifikate für Konnektoren ausstellen dürfen (siehe [TIP1-A_3937]).~~

~~**TIP1-A_3936 - Durchsuchen CA-Zertifikate**~~
~~Das eHealth-Kartenterminal MUSS für die Prüfung gemäß [TIP1-A_3933] die Liste der im eHealth-Kartenterminal gespeicherten CA-Zertifikate durchsuchen. <=~~

~~**TIP1-A_3937 - Einbringen CA-Zertifikate**~~
~~Der Hersteller des eHealth-Kartenterminals MUSS im Fall, dass [TIP-A_4115] nicht technisch im Kartenterminal umgesetzt wird, durch organisatorische Maßnahmen sicherstellen, dass nur für solche CAs die CA-Zertifikate in das eHealth-Kartenterminal~~

~~eingebracht werden, die auch tatsächlich Komponentenzertifikate für Konnektoren ausstellen dürfen.~~
~~Das Komponentenzertifikat des Konnektors wird durch das eHealth-Kartenterminal nur dann akzeptiert, falls alle Schritte ohne Fehler durchgeführt werden können.~~

~~**TIP1-A_3095-01 - SICCT-spezifischer TLS-Kanals, irrelevante Client-Authentisierung**~~
~~Das eHealth-Kartenterminal MUSS den TLS-Verbindungsaufbau akzeptieren unabhängig davon, ob der Client sich erfolgreich als Konnektor authentisierte oder nicht. <=~~

~~*Hinweis: Gründe für eine erfolglose Client-Authentisierung als Konnektor sind beispielsweise:*~~

1. ~~*Fehlendes oder fehlerhaftes Authentisierungszertifikat des Clients,*~~
2. ~~*End-Entity-Zertifikat zeigt nicht die Rolle "Konnektor",*~~
3. ~~*fehlende oder fehlerhafte Signatur im Rahmen der Client-Authentisierung.*~~

3.11 Zustandsautomat für den SICCT spezifischen TLS-Kanal

A_22456 - SICCT-spezifischer TLS-Kanal, Zustandsautomat
Das eHealth-Kartenterminal MUSS sich für den SICCT-spezifischen TLS-Kanal gemäß dem folgenden Zustandsautomaten verhalten:

1. *State_NoSicctTls*:
   Dies ist der Ausgangszustand eines eHealth-Kartenterminals bevor ein SICCT-spezifischer TLS-Kanal aufgebaut wird und der Endzustand nach Abbau eines SICCT-spezifischen TLS-Kanals.
2. *State_InvalidClient*:
   Aus dem Zustand *State_NoSicctTls* wird dieser Zustand erreicht, wenn ein TLS-Client den TLS-Handshake für eine SICCT-spezifische TLS-Verbindung erfolgreich durchläuft, aber die Client-Authentisierung als Konnektor erfolglos war. Durch Abbau des SICCT-spezifischen TLS-Kanals geht dieser Zustand nach *State_NoSicctTls* über.
3. *State_ClientWithoutPairing*:
   Aus dem Zustand *State_NoSicctTls* wird dieser Zustand erreicht, wenn ein TLS-Client den TLS-Handshake für eine SICCT-spezifische TLS-Verbindung erfolgreich durchläuft und sich erfolgreich als Konnektor authentisiert, aber der dabei präsentierte öffentliche Schlüssel aus dem Client-Zertifikat in keinem Pairing-Block enthalten ist.
   a. Durch Abbau des SICCT-spezifischen TLS-Kanals geht dieser Zustand nach *State_NoSicctTls* über.
   b. Dieser Zustand geht nach *State_ClientWithPairing* über, wenn der öffentliche Schlüssel aus dem Client-Zertifikat in einen Pairing-Block eingetragen wird (initiales Pairing, oder Wartungs-Pairing, siehe dort).
4. *State_ClientWithPairing*:
   a. Aus dem Zustand *State_ClientWithoutPairing* wird dieser Zustand wie im vorherigen Punkt beschrieben erreicht.
   b. Aus dem Zustand *State_NoSicctTls* wird dieser Zustand erreicht, wenn ein TLS-Client den TLS-Handshake für eine SICCT-spezifische TLS-Verbindung erfolgreich durchläuft und sich erfolgreich als Konnektor authentisiert und der

dabei präsentierte öffentliche Schlüssel aus dem Client-Zertifikat ist in einem Pairing-Block enthalten.

c. Durch Abbau des SICCT-spezifischen TLS-Kanals geht dieser Zustand nach *State_NoSicctTls* über.

[<=]

Hinweis: Der im Folgenden spezifizierte Status-Code SW1SW2 = '6982' = SecurityStatusNotSatisfied wird weder in [SICCT] noch in diesem Dokument als Antwort auf ein SICCT- oder EHEALTH-Kommando verwendet.

A_22461 - Erlaubte Kommandos für State_NoSicctTls
Im Zustand *State_NoSicctTls* DARF das eHealth-Kartenterminal WEDER SICCT- NOCH EHEALTH-Kommandos ausführen.**[<=]**

TIP1-A_3136-01 - Erlaubte Kommandos für State_InvalidClient
Im Zustand *State_InvalidClient* MUSS das eHealth-Kartenterminal

1. die in CMD_KT_0004 gelisteten Kommandos regulär ausführen falls diese über den SICCT-spezifischen TLS-Kanal empfangen wurden und
2. alle übrigen SICCT-Kommandos mit dem Status-Code SW1SW2 = '6900' = CommandNotAllowed ablehnen und
3. alle EHEALTH-Kommandos mit dem Status-Code SW1SW2 = '6901' ablehnen.

[<=]

Tabelle 2: CMD_KT_004, Liste ausführbarer Kommandos im Zustand State_InvalidClient

Kommandobezeichner
SICCT CT INIT CT SESSION
SICCT CT CLOSE CT SESSION
SICCT GET STATUS
SICCT SET STATUS
SICCT CT DOWNLOAD INIT
SICCT CT DOWNLOAD DATA
SICCT CT DOWNLOAD FINISH

TIP1-A_3096-01 - Erlaubte Kommandos für State_ClientWithoutPairing
Im Zustand *State_ClientWithoutPairing* MUSS das eHealth-Kartenterminal

1. die in CMD_KT_0004 und CMD_KT_0005 gelisteten Kommandos regulär ausführen falls diese über den SICCT-spezifischen TLS-Kanal empfangen wurden und
2. alle übrigen SICCT- und EHEALTH-Kommandos mit dem Status-Code SW1SW2 = '6900' = CommandNotAllowed ablehnen.

[<=]

Tabelle 3: CMD_KT_005, Liste ausführbarer Kommandos im Zustand State_ClientWithoutPairing

Kommandobezeichner
Kommandos aus Tabelle CMD_KT_004
EHEALTH TERMINAL AUTHENTICATE

TIP1-A_3097-01 - Erlaubte Kommandos für State_ClientWithPairing
Im Zustand *State_ClientWithPairing* MUSS das eHealth-Kartenterminal alle SICCT- und EHEALTH-Kommandos regulär ausführen falls diese über den SICCT-spezifischen TLS-Kanal empfangen wurden.**[<=]**

TIP1-A_3266-01 - Kartenkommandos für State_ClientWithPairing
Das eHealth-Kartenterminal MUSS alle ISO-7816 APDUs (siehe Envelope in [SICCT])

1. im Zustand *State_ClientWithPairing* regulär ausführen.
2. mit dem Status-Code SW1SW2 = '6900' = CommandNotAllowed ablehnen, wenn es nicht im Zustand*State_ClientWithPairing* ist.

[<=]

Hinweis: Der Status-Code SW1SW2 = '6900' = CommandNotAllowed wird in [gemSpec_COS] nicht verwendet.

~~In dieser Phase wird das korrekte Shared Secret (ShS.KT.AUT) nur durch den Konnektor geprüft. (Durch einen folgenden Aufruf von EHEALH TERMINAL AUTHENTICATE mit P2=02 gemäß Abschnitt 2.5.2.2). Das KT selbst bleibt passiv.~~

~~Damit der Konnektor die KT-Identität überprüfen kann, präsentiert das Terminal sein SMKT-Zertifikat (C.SMKT.AUT) dem Client im Rahmen des TLS-Verbindungsaufbaus. Der Konnektor prüft, ob es sich um ein gültiges SMKT-Komponentenzertifikat handelt und ob ihm das vom Kartenterminal präsentierte Zertifikat durch ein Pairing bekannt gemacht wurde. Handelt es sich nicht um ein gültiges SMKT-Komponentenzertifikat, wird der TLS-Verbindungsaufbau abgebrochen. Ist das Zertifikat ein gültiges SMKT-Komponentenzertifikat welches jedoch noch nicht mittels Pairing am Konnektor bekannt gemacht wurde, akzeptiert der Konnektor die TLS-Verbindung, jedoch stuft er das Kartenterminal als nicht vertrauenswürdig ein und führt nur jene SICCT- und EHEALTH-Kommandos aus, die in Kapitel 3.11.2 angeführt sind. Sind beide Prüfungen erfolgreich, wird die TLS-Verbindung akzeptiert. Der TLS-Verbindungsaufbau ist nach diesem Schritt abgeschlossen.~~

~~Ist für das Kartenterminalzertifikat am Konnektor Pairing-Information vorhanden, so prüft der Konnektor nach erfolgtem TLS-Aufbau die Pairing-Information (siehe Kapitel 2.5.2.2). Schlägt diese Prüfung fehl, wird die Verbindung abgebrochen.~~

~~3.11.1 Positivliste für Kommandos ohne gültiges Konnektorzertifikat~~

~~Unabhängig vom Stand des Pairings (siehe dazu Kap. 2.5.2) und unabhängig vom während des TLS-Verbindungsaufbaus vom Client präsentierten Zertifikat muss es am Kartenterminal möglich sein, ein Firmware Update zu ermöglichen und Statusinformationen abzufragen.~~

~~TIP1-A_3137 - „Liste ausführbarer Kommandos ohne gültiges Konnektorzertifikat“~~
~~Das eHealth-Kartenterminal MUSS nach dem TLS-Verbindungsaufbau unabhängig vom Stand des Pairings und unabhängig vom während des TLS-Verbindungsaufbaus vom Client präsentierten Zertifikats am Kartenterminal die in [gemSpec_KT#CMD_KT_0004] „Liste ausführbarer Kommandos ohne gültiges Konnektor-zertifikat" gelisteten Kommandos ausführen können. <=~~

~~Andere SICCT- oder EHEALTH-Kommandos als die in Tabelle 18 gelisteten Kommandos dürfen nicht ausgeführt werden, falls es sich bei dem zum TLS-Verbindungsaufbau präsentierten Clientzertifikat um kein gültiges Konnektorzertifikat handelt (siehe [TIP1-A_3136]).~~

~~3.11.2 Positivliste für Kommandos ohne gültige Pairing-Information~~

~~Unabhängig vom Stand des Pairings (siehe dazu Kap. 2.5.2) und unabhängig vom während des TLS-Verbindungsaufbaus vom Client präsentierten Zertifikat muss es möglich sein, das Kartenterminal in Betrieb zu nehmen.~~

~~TIP1-A_3098 - Aufbau des SICCT-spezifischen TLS-Kanals, zusätzlich erlaubtes Kommando bei gültigem Konnektorzertifikat ohne Pairing~~
~~Das eHealth-Kartenterminal MUSS zusätzlich zu den in [gemSpec_KT#CMD_KT_0004] gelisteten Kommandos auch die in [gemSpec_KT#CMD_KT_0005] gelisteten Kommandos unabhängig vom Stand des Pairings am Kartenterminal zur Ausführung anbieten, wenn es sich beim während des Aufbaus des SICCT-spezifischen TLS-Kanals vom Client präsentierten Zertifikat um ein gültiges Konnektorzertifikat handelt. <=~~

~~Andere SICCT- oder EHEALTH-Kommandos als jene in Tabelle 19 sowie in Tabelle 18 (siehe Kapitel 3.11.1) aufgeführten dürfen nicht ausgeführt werden, falls der öffentliche Schlüssel des beim TLS-Verbindungsaufbau präsentierten Konnektorzertifikats nicht in den Pairing-Informationen des Kartenterminals enthalten ist (siehe [TIP1-A_3096]).~~

3 Änderungen an Produkt- und Anbietertypsteckbriefen

Für die oben genannten Festlegungen werden die nachstehenden Prüfverfahren im Produkttyp-Steckbrief [gemProdT_KT] definiert.

Id	Titel	Produkttyp	Prüfverfahren
TIP1-A_3255	CA-Zertifikate der relevanten TSP speichern	KT	Sich.techn. Eignung: CC-Evaluierung, funkt. Eignung: Test Produkt/FA
TIP1-A_3256	CA-Zertifikate in Kartenterminal und anschließende Speicherung	KT	Sich.techn. Eignung: CC-Evaluierung
TIP1-A_3257	Schutz CA-Zertifikate	KT	Sich.techn. Eignung: CC-Evaluierung
TIP1-A_6482	Anzahl CA-Zertifikate	KT	funkt. Eignung: Herstellererklärung
TIP1-A_3094	Aktualisierung von CA-Zertifikaten der Komponenten-PKI	KT	Sich.techn. Eignung: CC-Evaluierung
TIP1-A_3941	Update von TSP-Zertifikaten	KT	Sich.techn. Eignung: CC-Evaluierung
TIP1-A_3158	TSP-Update-Mechanismus	KT	Sich.techn. Eignung: CC-Evaluierung, funkt. Eignung: Test Produkt/FA
TIP1-A_3159	TSP-Update-Mechanismus für KT ohne Firmware-Update	KT	Sich.techn. Eignung: CC-Evaluierung, funkt. Eignung: Test Produkt/FA
A_22888	K_KT, CA-Zertifikat suchen	KT	funkt. Eignung: Test Produkt/FA
A_22889	K_Hersteller-KT, Liste zulässiger CA-Zertifikate	KT	Sich.techn. Eignung: CC-Evaluierung, funkt. Eignung: Test Produkt/FA
A_22890	K_KT, erfolgreicher TLS-Handshake	KT	funkt. Eignung: Test Produkt/FA
A_22456	SICCT-spezifischer TLS-Kanal, Zustandsautomat	KT	funkt. Eignung: Test Produkt/FA
A_22461	Erlaubte Kommandos für State_NoSicctTls	KT	funkt. Eignung: Test Produkt/FA
TIP1-A_3136-01	Erlaubte Kommandos für State_InvalidClient	KT	funkt. Eignung: Test Produkt/FA
TIP1-A_3096-01	Erlaubte Kommandos für State_ClientWithoutPairing	KT	funkt. Eignung: Test Produkt/FA

© gematik – öffentlich_Entwurf

TIP1-A_3097-01	Erlaubte Kommandos für State_ClientWithPairing	KT	funkt. Eignung: Test Produkt/FA
TIP1-A_3266-01	Kartenkommandos für State_ClientWithPairing	KT	funkt. Eignung: Test Produkt/FA

518